CROSSTALK

Stories of Change

Report Documentation Page

1. REPORT DATE **AUG 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**
4. TITLE AND SUBTITLE **CrossTalk: The Journal of Defense Software Engineering. Volume 20, Number 8, August 2007**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **OO-ALC/MASE,6022 Fir Ave,Hill AFB,UT,84056-5820**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**32**	

Stories of Change

Software Engineering Technology

CROSSTALK

Departments

ON THE COVER
Cover Design by Kent Bingham

Correction: Due to a printer error, inaccuracies were introduced into the July Table of Contents. Please see our Web site for the correct Table of Contents <www.stsc.hill.af.mil/crosstalk/2007/07>.

The Right Way to Change

Change has always been part of life, yet many still have difficulty with this reality. Difficulty accepting change may stem from a simple desire to stay within a comfort zone or the presence of a turbulent history of failed attempts. If the latter is the case with organizational change, any new attempts at change must account for the historical obstacles that arose with the aim of creating plans for a new change process that addresses these obstacles. While identifying and planning on how to address obstacles to change, some considerations should include preparations for any required resources, ensuring sponsorship and addressing resistance.

Conversely, if an organization has a history of successfully implementing change, there is good reason to be optimistic that new changes will also be successful, as long as the past processes are leveraged. Regardless, organizations should only undergo change when the reasons and timing are right and the proper tool set is in place to make change successful.

This month we start with a story of successful change in Iraq. In *Good News From Iraq*, CAPT Steven Lucks (Ret.) discusses changing the infrastructure of Iraq with the help of developing a data control system that requires software, computers, supporting electronics, and participation from multiple organizations in multiple countries. Next, Nelson Perez and Earnest Ambrose relate their story of successful software process improvement in *Reaching Maturity Level 2/Capability Level 3 in Nine Months*. We conclude our theme article section with Deb Jacobs' ideas for controlling change in *Controlling Organizational Change: Beyond the Nightmare*. In her article, she proposes several out-of-the-box ideas to implement change.

As we move to our supporting sections, Harvey Reed and COL Fred Stein (Ret.) introduce net-centric conversations as a way to track agility information among software systems in *Net-Centric Conversations: the Enterprise Unit of Work*. In *A Unified Service Description for the Global Information Grid*, Dr. Yun-Tung Lau identifies linkages between existing service description standards used within the Global Information Grid (GIG) and Department of Defense frameworks with the intent to provide an end-to-end picture of a service module and its role in a GIG enterprise. Next, Joe Schofield discusses a process to estimate the number of latent defects remaining in software. His discussion in *Beyond Defect Removal: Latent Defect Estimation With Capture-Recapture Method* can be leveraged to decide how to proceed with a software product, including planning for rework.

With any change, there will be impact and, as a result, some resistance and decline in productivity. As we change, let's do it for the right reasons. Consider a few things before you implement any change to your organization.
- What will be the pros and cons if the change is or is not implemented? Are the pros worth the cost?
- How many people will be impacted by the change? How many of them will even comply with the change? How long will it take them to adjust to the change?
- How long will you be in your position? Will your replacement just change everything back again? Will the employees change everything back again even before your replacement has a chance to?

When we implement change, let's take time to weigh the value of the change against the headaches and problems it may cause, then try to do what really is best.

Norman R. LeClair

Norman R. LeClair
Ogden Air Logistics Center, Co-Sponsor

Good News From Iraq

CAPT Steven J. Lucks (Ret.)
U.S. Navy

Building a data center in a war zone is an extreme challenge requiring creativity, diplomacy, statesmanship, and the can-do spirit. This is the story of an Iraqi and American mixed team that, with uncommon persistence and under extreme duress, built a world class data center and fully functioning office complex.

The rebuilding of Iraq effort, which was funded by the United States Congress in 2003, allocated about $18 billion for Iraqi reconstruction and aid. Of that, about $7 million went to funding the building of a data center. In addition to software, the entire system included the buildings, air conditioning, elevators, office furniture, electricity; and the infrastructure for all the sites that needed the information. This was not a typical data center building project like one in the United States; this literally started with nothing.

"It won't do any good to build facilities if they can't be managed," said Dennis Plockmeyer, a retired Navy Construction Battalion Captain, and now, the Chief Information Officer for the Project Contracting Office Iraq, which oversees logistics for all of Iraq's $18 billion reconstruction initiatives.

Plockmeyer had been in Iraq since September 2003 and in Baghdad's green zone, a section of the city from which the coalition forces managed their major reconstruction efforts. I, a Navy Surface Warfare Officer Captain, had been in Iraq since December 2003 and had worked in and around Baghdad and other key cities for the Coalition Provisional Authority before joining Plockmeyer's team in July of 2004 as his Operations Director. We both served the Department of Defense (DoD) as senior civilians.

At the heart of the data center building plan was an effort to introduce an asset-management system to Iraqi public officials who, in many cases, had never used anything more than pencil and paper to manage vital national assets. "It doesn't do any good if you build all of these facilities and then walk off without giving the recipients the tools and the wherewithal to manage them," said Plockmeyer.

Problems and Issues

The Information Technology (IT) team, which consisted of contractors from the United States, including the native small business association firms, and local Iraqis, could have built an IT system to solely run the coalition's reconstruction effort. That would have been cheaper and easier, since it would function entirely in English and run on off-the-shelf and DoD-supplied software. Instead, they opted for the complexity of writing additional code that let the system run in parallel with Arabic and Kurdish. This option ensured that the investment in technology and processes needed to manage the reconstruction had ongoing value that could be transferred to the Iraqis, focusing on what happens the day after the contractors leave. The master database built by the combined team was named the Iraq Reconstruction Management System (IRMS).

The major components of the IRMS system included Maximo (owns the requirements/assets); ESRI (defines the location); Oracle e-Business (exhibits cost and performance), Primavera P3ec (develops the schedule), DoD standard procurement system (authors the contracts), DoD Corps of Engineers (ACE) financial management system (manages the finances), DoD requirements management system (captures the construction), Oracle e-Success (delivers the estimates), Expedition (provides project controls), and Oracle Portal (spans the program, gateway to the solution) all running on Unix, Linux, and Microsoft (MS). Net operating systems were accessed via MS Office on the desktop. Connecting the various components that comprise the system was relatively easy compared to the logistics and danger to workers building the data center and offices. Regarding the software *build*, the distance and time zone differences had to be taken into consideration because Iraq as well as Virginia, California, and Washington had to be linked and functioning in real time. Personnel in Iraq often worked 18 hours a day, seven days a week in the software effort. Configuration management was a central issue to ensure success.

Harder to accomplish than building the software was building the data center and its infrastructure. Many of the Iraqi men had limited education due to what Iraqis reported as Saddam Hussein's tendency to restrict education for the males to the sixth grade. This made it difficult because the team had to find qualified locals who turned out to be educated females. This presented a problem in a culture dominated by men where women were not valued for their knowledge or ability to work outside of the home. Overcoming these cultural differences by use of relationship management, statesmanship, diplomacy, and trust building allowed the formation of a world-class team.

By working with the Iraqi Console for Employment, the project received a steady flow of resumés from young Iraqi men and women who wanted to participate in what they called a *privilege to work* environment. There were many technologically literate Iraqis anxious to apply their skills to the rebuilding effort. They understood their skills might not be the most current, but they were ready to learn. While few of the workers had worked with advanced applications such as Maximo, many had basic technology skills and were familiar with Oracle and other common IT environments. The issue of training and mentoring the basics of Software Engineering Institute/Capability Maturity Model® Integration and Computer Society for Software Engineering by the Institute of Electrical and Electronics Engineers, Inc. for the software teams posed little problems in understanding by the Iraqis. However, using Project Management Institute concepts for the teams that were involved with the physical building and plant layout was one of the hardest things to do because most Iraqis and some contractors in the building trades knew very little of how projects needed to be executed using a repeatable method.

For example, the simple idea of grounding the data center and all the sys-

tems was not something understood by all and had to be explained to both contractors and Iraqis alike. Other building and infrastructure issues were getting all the requirements in for electricity, network, phones, televisions, contractor housing, and hospital needs. This was needed because the digging and building of a conduit to accommodate these different needs had to be planned, the correct fiber-optics had to be ordered, and construction had to be carefully timed because of security issues with Iraqi construction contractors. We also encountered problems with the electrical system used in Iraq. The Iraqi system is based on the British electrical system, and American companies shipped a U.S.-based electrically supported system. Because everything had to be flown in, *work-arounds* had be put into place until the correct equipment was shipped. For one piece of equipment, the system had to be rewired because replacing the equipment would have cost more than the rework. Another issue that had to be overcome was the heat, sand, and dust. In the direct sunlight on top of the building, temperatures reached 160 degrees Fahrenheit and melted the equipment used to communicate with the satellites.

The building that was used to house the data center was originally built by Saddam's sons and called the *Hall of Records and Justice*. This building stored millions of records detailing all the people Saddam's regime had murdered; many were tortured in the main square under it. The data center refurbishment and set-up required that personnel hand-carry every desk, chair, individual computer, phone, light, and other office equipment to fill the seven-story building, and then to build the data center, they had to hand-carry all 110 servers and related hardware up seven floors to make the system work. This was done without the aid of air conditioning or elevators in temperatures of 130 to 140 degrees, but there was a real sense of ownership and no complaints about the unusually harsh working conditions. What made it more difficult than accomplishing anything in the western world was that the Iraqis were constantly being threatened while coming and going to and from their work centers. At times, safe rooms had to be set up so that the workers could stay overnight.

Plockmeyer and I created a work environment that encouraged trust and creative thinking and maintained focus, intensity, and persistence. Even under severe wartime work conditions, we took the teams out to dinner and set up a small movie theater inside the building where they could stay and be somewhat safe. In turn, the Iraqis brought local food and shared their family cooking.

Security Issues

To help contractors understand that working in Iraq was not like working *back home*, training on cyber security for all users had to be accomplished. The team used computer forensics to track users who tried to violate the rules. For example, a problem that had to be overcome was that contractors tried to send sensitive information back to the United States, which could have put them or the Iraqi workers in grave danger because the information was not encrypted when transmitted. The ability to bind security systems to the physical systems within the main computer center operations area was developed so that all workers could feel safer.

"It doesn't do any good if you build all of these facilities and then walk off without giving the recipients the tools and the wherewithal to manage them."

Another challenge that had to be overcome was that the system interfaced with the State Department, the ACE Gulf Region division (GRD), the coalition, and the Iraqi government. The team was instrumental in resolving the information assurance challenges inherent in migrating from a military to a commercial environment while preserving the warfighters' network and accommodating and developing secure systems (including Top Secret and higher security levels) for the military to be used in the same building as Iraqis. This effort included the development of Voice-over Internet Protocol and wireless (Wi-Fi) systems (both secure and commercial), keeping a defense in-depth philosophy so that data (both voice and computer-generated) would not compromise the organizations that needed the information. The team also supported diverse needs of multiple, direct-support entities and ensured that the IT infrastructure accommodated six different networks without compromising information security or system capability.

Building a System That Would Work for Iraq

Plockmeyer focused on making sure that modules could be added that would monitor the health of oil pipelines and would alert authorities to a drop in pressure caused by mechanical failure or sabotage. The coalition's asset-management system also was able to capture data from remote diagnostic and management technologies being built in some of the newer Iraqi buildings. Plockmeyer said that some of the construction blueprints he had seen called for utility plants to incorporate advanced supervisory control and data acquisition technologies – a first in Iraq.

Coalition officials wanted to introduce the asset-management system to Iraqi administrators in small doses. For example, the system was built to manage the building of the electricity sector around Baghdad and then later to all of Iraq.

After four years, Plockmeyer and I believe the progress the coalition made in Iraq has been largely obscured by news that focuses mostly on the day-to-day violence. The list of projects completed or initiated under the coalition's watch – and managed through the asset-management system – is lengthy. Each week, about $75 million in new construction work begins on projects ranging from water-treatment and waste-management systems to new schools.

Ever-present in a war zone like Iraq was the threat of attacks on coalition personnel and any Iraqis working with them. Even from the living quarters, personnel could hear and feel the rockets and mortar shells that Iraqi insurgents occasionally fired into the green zone. The violence did not delay the implementation of the core asset-management system. Plockmeyer said the following about my work:

> Lucks made sure that the Internet access was widely available so that the modules were fully utilized by some of the more far-flung Iraqi ministry outposts and saved $2 million in operating expenses.

U.S. Government Makes IRMS the Standard

An interagency Information Technology Working Group (ITWG) was formed in August 2004 with the mandate to consolidate all U.S. government-funded and managed relief and reconstruction project information across all sectors and organizations throughout Iraq into one database for reporting to the U.S. Congress through the U.S. Ambassador to Iraq and the

COMING EVENTS

COMING EVENTS: **Please submit coming events that are of interest to our readers at least 90 days before registration. E-mail announcements to: nicole.kentta@hill.af.mil.**

Commander, Multi-National Force-Iraq. As planned, the U.S.-based team, along with Iraqi citizens, implemented the asset-management system at various Iraqi ministries. "Working shoulder to shoulder on the same system gives you the basis for a successful turnover," Plockmeyer said.

By leveraging the same IT system already in use by the Project and Contracting Office, other U.S. agencies benefit from the enterprise network with little or no capital investment, according to the ACE. The master database built by the joint American and Iraqi team, the IRMS, was the system chosen by the Iraq Reconstruction Management Office (IRMO).

The IRMO chair of the ITWG and the director of the Primary Control Officer/GRD National Reconstruction Operations Center have championed IRMS as the interagency solution not only for reporting the total U.S. government effort but also for providing multinational forces – integrated field commanders with situational awareness of relief and reconstruction efforts in their areas of operation, allowing for greater synchronization of efforts.

According to the ACE, as of May 2007, IRMS will be turned over to the Iraqi government as an archive of the total U.S. government effort, which will help in its budgeting for operations and maintenance of new facilities and future master planning.

Summary

Building the data center system involved many obstacles, some of which hopefully are not faced during the development of most systems: addressing the requirements of others that would want to access this system in addition to our own requirements, danger of attack on those developing the system, cultural adversity of men and women working together, limited skills with commercial off-the-shelf software used, electrical inconsistencies, and other extreme working conditions. These were overcome with relationship management, statesmanship, diplomacy, trust building, technical training, security, dedication, and perseverance.

The IT effort in Iraq was an Iraqi and American team effort that has benefited contractors, the coalition, and Iraq and has helped facilitate positive development throughout that country.◆

About the Author

CAPT Steven Lucks (Ret.) served with distinction for 30 years in the Navy Reserves, the latest being in Iraq. He currently is an independent consultant working on issues dealing with Agile software development, service-oriented architecture risks, the Health Insurance Portability and Accountability Act, security, and e-discovery.

Phone: (208) 523.6273
Cell: (208) 521.1980
E-mail: luckssj@msn.com

LETTER TO THE EDITOR

Dear CROSSTALK Editor,
I am writing in regards to the Sponsor's Note by Kevin Stamey titled *Lead, Follow, or Get Out of the Way* in the April 2007 issue of CROSSTALK.

I have heard this expression so many times and it drives me crazy to hear it spoken, as I would claim, improperly. I don't know that Lee Iacocca did not actually say *lead or follow, BUT get out of the way*, but I am sure that is what he meant.

Too many times people and organizations stand in the middle of the road drawing a crowd, talking the talk, taking the focus, taking the credit, promising the world, and churning out reworked platitudes. Leadership means knowing where the pack should go and having the right stuff to pull them there.

There is nothing wrong with following, of course, because without actually implementing the plans of leaders, we would have no progress. So I would say to the talking heads, lead with insight and wisdom, or follow with respectful allegiance, but do not just stand there. Drawing a crowd causes a distraction.

So, over my desk is MY version of the expression:

Lead or follow, but get out of the way!

– Julian Opificius
Software Engineering Manager
Shadin Avionics
<julian.opificius@shadin.com>

Lessons Learned in Using Agile Methods for Process Improvement

Nelson Perez
Sierra's Edge, Inc.

Ernest Ambrose
MORI Associates, Inc.

This article presents lessons learned from a process improvement (PI) effort that took an organization from no formal process capability to the implementation of the Software Engineering Institute (SEI) Capability Maturity Model Integration (CMMI®) using the continuous representation with a focus on the staged representation's Maturity Level 2 (ML2) process areas (PAs). This article summarizes techniques that were used to reduce the overall time to achieve institutionalization of new processes as well as what worked and what could be further improved.

When an organization decides to newly embark on PI, there are several issues that influence the amount of effort involved and the effective timeline to achieve a particular PI goal. Lessons gleaned from the software development world in the use of incremental or iterative approaches can be applied to any type of project to achieve similar results, including PI. With proper planning, the end goal can be reached in a greatly accelerated fashion. Effective planning is not the only element, however, that should be considered when reducing duration or budget.

This article examines the approach taken at MORI Associates on a PI effort that not only met its goals but exceeded the expectations of all involved. With about 75 employees spread across seven projects, MORI provides information technology, engineering, and operations services for government agencies and private industry. Included herein are some of the techniques employed and lessons learned along the way.

Be Prepared to Make a Significant Commitment

Before we examine methods to reduce effort and duration, we should discuss the costs and impact involved in a PI effort. Depending on the amount of new processes involved, there can be a considerable amount of effort required on the part of management, project staff, and the overall organization. This commitment will start with the PI effort planning stage, increase substantially as the projects implement new processes, and produce new and potentially large and unexpected work products (e.g., requirements documents and requirements traceability matrices) and will continue even after the appraisal as these processes become part of the new way of doing business.

At MORI, the organization was fully committed to the change process. This commitment began at the top with the sponsor, President/Chief Executive Office of MORI, Shahnaz Deldjoubar, and continued through upper management and out to the staff. The sponsor had all projects perform an in-depth analysis of the impact to effort, resources, and schedule. Their highest priority was their established commitments to their customers in terms of agreed-upon deliveries, services, and schedules. The projects were able to update their plans to implement the new processes without impacting their customer commitments. Along the way, the staff also contributed some of their personal time, such as conducting software engineering process group (SEPG) meetings during lunch and attending after-hours training sessions. The areas that involved the greatest effort were requirements documentation and traceability, configuration management, and project planning and monitoring.

A summary of the effort involved for process development is shown in Table 1, while the impact felt after process rollout is shown in Table 2.

Table 1: *Process Development Effort*

Activity	Responsible Party	Effort Hours
Develop Processes, Policies, and Work Product Templates	Consultant	647
Meetings, Process Changes, and SEPG Bootstrap	Consultant	186
Review, Approve, and Revise Process Assets	SEPG, Steering Committee, Sponsor	404
Total		1237

Table 2: *Process Implementation Effort for Projects and Organization*

Activity	Responsible Party	Effort Hours
Training and Mentoring	Consultant	167
Training and Mentoring	Organization	251
Implement New Project-Specific Processes (38 work products)	Projects	724*
Implement New Organizational Processes (8 work products)	SEPG, Steering Committee, Sponsor	54

* Effort represents average per project over a nine-month window for maintaing Web-enabled Management Information Systems.

® CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

In comparing this consultant-centric and Agile-based approach to what is expected by an established CMMI estimation model, it is 30 percent more efficient than the most optimistic CMMI process development estimate [1, 2].

Outsource Process Development to Reduce Impacts to Staff

Enlisting a consultant to develop your process assets can be helpful when the capability or PI experience is lacking or the staff is too busy to devote the time needed to develop process assets. At MORI, all process assets were developed by an external consultant with each PA requiring an average of 80 consulting hours to develop, including 77 hours to plan the effort. Consulting time was divided among process development (65 percent); mentoring and training (16.5 percent); and meetings, action items and PIs (18.5 percent). With only a 65 percent availability to develop processes, a four-month calendar (assuming 159 man hours per calendar month) effort required about six months. In the absence of a full-time consultant, companies might assign one or more persons to each PA. Assuming a staff of seven (i.e., one person per PA) and a 20 percent availability (eight hours per week), it should take about three months to develop all the processes. However, most staff assigned to process development tend to be pulled off to perform their normal responsibilities. Availability usually shrinks to 5 percent (two hours per week) and sometimes to zero for extended lengths of time. At 5 percent, process development with a staff of seven can stretch from three months to as many as 12 months or more. With a consultant, the process development schedule can become more deterministic and the staff can stay focused on their projects and the new effort involved. At MORI, the five-person SEPG committed about 8 percent of their time in support of the process development effort and the percentage of time rises to about 13 percent when you include training and mentoring activities.

Run the CMMI Effort as an Agile Project and Use It to Pilot Key Concepts and Tools

Regardless of who is developing the process assets, start prototyping processes from the very beginning by treating the CMMI effort as a pilot project, experimenting with processes that can be adapted and eventually transitioned to the organization.

Our PI effort planned on implementing an incremental development model but ended up implementing an incremental/iterative model. The process development sequence was planned to ensure that long lead items would be kicked off first (in our case documenting requirements and creating a requirements traceability matrix for each project) followed in importance by what seemed like a logical order based on a typical development life cycle. The approach used the continuous representation of the CMMI model. We chose a target profile of the staged representation's ML2 PAs at Capability Level 3 (CL3), Decision Analysis and Resolution (DAR) at CL3, and Organizational Process Definition (OPD) and Organizational Process Focus (OPF) at CL1. We also created a template to document project requirements

> *"Regardless of who is developing the process assets, start prototyping processes from the very beginning by treating the CMMI effort as a pilot project, experimenting with processes that can be adapted and eventually transitioned to the organization."*

for the Requirements Development (RD) PA. Supplier Agreement Management was deemed not applicable (N/A) and confirmed by our lead appraiser. Although a continuous representation was chosen to execute the PI project, the actual goal was to achieve a staged representation ML2 rating.

The planned order of process development was the following:
1. OPD, OPF, DAR.
2. Requirements Management (REQM) + Requirements Template.
3. Project Planning (PP).
4. Project Monitoring and Control (PMC).
5. Configuration Management (CM).
6. Process and Product Quality Assurance (PPQA).
7. Measurements and Analysis (MA).

Although this order was generally followed, some of the processes and associated work products were actually created and released in an iterative manner while others were developed out of cycle, as some portions of their policies and work products were prototyped for use by the SEPG, followed by pilots on select projects and further iterated on as feedback was generated.

In running the effort as a project, reports on project progress, risks, and issues should be made to the organization's upper management, including the sponsor, on a periodic basis. In our case, a monthly project management review (PMR) was implemented and a PMR slides template was developed. The PMR slides template was developed in an iterative fashion, as the effort progressed. Kicking off the PMR process with monthly reviews of the PI effort helps accomplish quite a few objectives. It communicates progress to the sponsor while planting the seeds of the new PMR process, familiarizing them with the format, and creating a more formal review process. Having the project leaders participate early on allows them to learn by example, even before the process has been documented.

Pilot Key Processes in an SEPG

Use the SEPG to prototype several of the high return on investment (ROI) processes and templates.

Prototype an action item management process and action item log template (PMC), create a process change management process and process change request (PCR) forms (OPF), define process standards (OPD), and define meeting agendas and minutes (PMC) which can be applied to many types of meetings. Once the SEPG processes have stabilized, tailor the process change management process and templates to suit the REQM and CM change control processes. Tailor the meeting agenda and minutes template to use for project staff and customer meetings, for the software configuration control board (CCB) and the requirements CCB. Define metrics to track project and SEPG activity. Discuss and track schedule progress and issues with the SEPG. Reuse these metrics to track project effort, schedule, and activities.

In our case, the project leads were so excited about having certain tools, especially an electronic way to track and manage action items, they piloted them without being asked, adapting them to suit their own needs well in advance of the

development of the associated work product templates. Their early adoption efforts helped identify issues and greatly reduced the PI effort duration and risk.

When asked for a list of the employees' most useful/important new tools and processes, the following is the feedback we received:

- Action Item Log.
- Weekly Status Meetings.
- PMR.
- Schedule Tracking.
- Change Control.
- Metrics and the project measurement repository.

Identify and Implement High-Impact Technology Improvements

During PI planning, determine if there are PAs that could benefit from the acquisition and integration of third-party support products to streamline what otherwise might be manually intensive processes. MORI decided to focus on acquiring tools to automate CM and defect tracking in the near term and possibly address requirements management in the long term.

To facilitate the acquisition process, develop and roll out a high-quality DAR process to the SEPG. MORI used a DAR process using an agreed upon set of evaluation criteria to acquire a freeware CM tool. Although the product review and selection process was detailed, thorough, and extensive, there were some unexpected issues that arose after the tool was installed.

Employ prototyping or simulation techniques when evaluating these critical products. Several issues with the CM system could have been avoided if we had, for example, prototyped check-in and check-out procedures for each candidate CM system solution. When we experienced these types of issues, we updated the DAR process (through a process change request and the SEPG) to identify more precise product evaluation criteria and incorporate simulation and prototyping as a requirement when selecting similar products. This lesson learned was then applied to the evaluation of the defect tracking system where simulation of the change state model was applied to the candidate products.

Leverage the Internet for Process Development Information

Leverage the resources of the Internet to

survey current industry for examples of policies, processes, work products, tools, and lessons learned. It is possible to benefit from the works of established processes, but approach with caution as not all examples will necessarily fit your organization. At MORI, the Internet was used to research example policies; earned value management, risk management, and example DAR processes; lessons learned templates; change request forms; configuration identification and naming conventions; and baseline tagging techniques.

Leverage Industry Standards

Use industry-accepted standards for documentation instead of creating your own. MORI purchased the Institute of Electrical and Electronics Engineers (IEEE) Software Engineering Collection

> *"The value of documentation standards such as those produced by the IEEE is that they are a result of the collaboration of many leading industry experts. By using such standards, you are leveraging a larger pool of expertise..."*

[3]. The value of documentation standards such as those produced by the IEEE is that they are a result of the collaboration of many leading industry experts. By using such standards, you are leveraging a larger pool of expertise rather than trying to come up with an internally produced standard, resulting in considerable savings.

Implement Defined Processes

Develop organizational standard processes (i.e. define critical processes at a CL3 level of detail) and tools that include tailoring guidelines instead of flowing down detailed process decisions to each project, as is the case for an ML2 organization (under the staged CMMI approach). This frees projects to do their work with-

out being encumbered with the need to become process experts, especially if they lack the capability to develop their own detailed processes, as is the case with many organizations just starting down the road of PI. Providing detailed PA process descriptions and procedures as well as standard forms, templates, and infrastructure (e.g. common project repository folder structure, CM library, and defect tracking tools) makes the job of project participants and upper management easier, especially when moving from project to project. It speeds institutionalization and simplifies the appraisal process. Processes were documented from the union of the classic IBM ETVX (entry, task, verification, exit) and Watts Humphrey's ETXM (entry, task, exit, measure) process architectures to yield an ETVXM (entry, task, verification, exit, measure) process architecture, where both measures and verification steps augment the description of the entry and exit criteria and tasks to be performed [4]. If going for CL3, remember to add explicit tailoring instructions to fully satisfy Generic Practice (GP) 3.1 and ensure the organization collects best practice examples for its process asset library.

Be CMMI Friendly

Make some of your processes and work products CMMI friendly and, hence, appraisal-friendly; show how they map to each PA. For example, in meeting agendas and minutes, create subsections for each PA. This will help guide important discussions while providing quite a bit of indirect evidence across several PA's. To simplify the appraisal, create project summary presentations that show how each PA is satisfied. Although the Standard CMMI Appraisal Method for Process Improvement (SCAMPI) Method Definition Document suggests creating presentations as a way to increase appraisal related oral affirmations [5], providing direct mappings to each PA within the presentation helps simplify the job of the appraisal team when it comes to the verification of objective evidence (of oral affirmations).

Outsource QA, Ensure Your Designated QA Lead Is Objective, and Keep QA Checklists Simple

An area that is often a challenge for most organizations new to process is QA. As Juran has noted, while companies are generally experts in their particular discipline such as product development, they
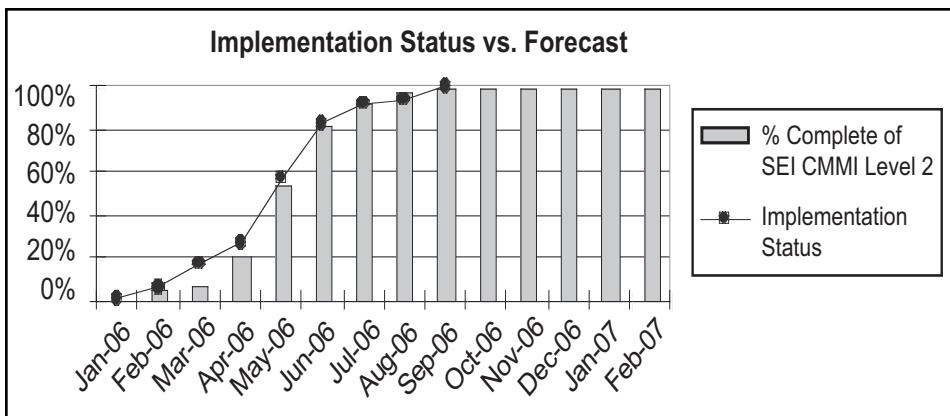
Figure 1: *MORI Associates' Process and CMMI Forecast and Compliance Profile*

"lack expertise in the 'quality disciplines' – the methodology, skills, and tools required to plan for quality" [6]. PPQA is a PA that is more open ended in terms of the details of its specific practices. Many companies find it a challenge to implement, and it is often found to be a weakness during appraisals. Common issues uncovered are that QA training is not properly addressed, that audits are not performed until right before the appraisal and thus not institutionalized, objectivity is not achieved, and audits of the audits are overlooked (i.e. GP2.9 for PPQA is not covered). This is typically the case for organizations lacking dedicated QA resources.

There are some short-term solutions: Assign an acting QA lead, distribute the audit responsibilities across the company, and consider outsourcing some of the audits. However, be careful about outsourcing QA. MORI learned that if QA audits are outsourced and the QA lead is not totally independent of the projects, it is necessary to have the consultant check back to ensure issues were addressed properly and in the appropriate timeframe. Also, make sure QA auditors are trained in the processes and work products they audit.

Create the Practice Implementation Indicator Database (PIID) Early and Get It Validated By a Competent Lead Appraiser

Create the PIID early and use it to track implementation status as you roll out the processes. Use a high quality lead appraiser to perform a gap analysis of the processes and validate the PIID mappings. Interpreting the model in the context of

> *"The overall forecast defines expected monthly process implementation goals for each PA (in terms of work product completion) and predicts the overall target date to reach full compliance with the CMMI model."*

many different approaches is a continual challenge. It is best to have an experienced set of eyes looking at the PIID.

Forecast and Track CMMI Compliance for the Life of Each Project and the Organization

Forecasting process implementation helps an organization track its progress and assess its appraisal readiness. As part of the sponsor's request to evaluate the

impacts to projects, the project leads worked with the PI consultant to negotiate the projected completion dates of each of the work products associated with the new processes. Using a simple spreadsheet-based tool, the projects and the organization were able to tie the compliance status of each of the specific and generic practices of each PA to the expected and actual completion dates of their associated work products. By initializing the tool with expected work product completion dates, monthly compliance goals were automatically generated.

This tool effectively creates a hybrid PIID that not only reports CMMI compliance but also allows projects to track the monthly status of the direct and indirect artifacts needed to satisfy the PA specific and generic practices. This hybrid PIID uses the work product status data entered in by the project leads to calculate a percentage of compliance for each PA and allows the project leads and organization to determine if they are meeting the planned forecast and still on track to achieve the overall PI effort as planned. This tool also generates an expected appraisal-readiness date for the PI effort and can be used as an input to revise the PI plan and schedule.

A good way to visualize this is through an example. Let us say that a specific practice requires four distinct work products to be generated in order for the practice to be fully implemented and therefore compliant with the CMMI. Let us also assume that each work product takes a month to create and is to be created in a serial fashion. The forecasted compliance trend would then be 25 percent, 50 percent, 75 percent, and 100 percent across this four month time period. One could then collect the actual status of each of these work products from each project as it progresses and average their statuses each month to visualize the organization's progress toward full compliance for the practice. For long-lead work products, such as the requirements traceability matrix (RTM), status tracking could be made more granular by reporting progress at the product component level, for example.

The overall forecast defines expected monthly process implementation goals for each PA (in terms of work product completion) and predicts the overall target date to reach full compliance with the CMMI model. This self-assessment also helps meet the requirements of OPF SP 1.2.

The forecast and achievement profile for process and CMMI compliance across the MORI organization is shown

Table 3: *Compliance of PAs Leading Up to Appraisal – Work Product Completion Perspective*

Process Areas	June 2006	July 2006	August 2006	September 2006
REQM	94%	98%	98%	100%
PP	97%	98%	100%	100%
PMC	93%	95%	96%	100%
CM	73%	84%	88%	100%
PPQA	75%	80%	84%	100%
MA	66%	96%	96%	100%

in Figure 1. The bars represent the forecasted process implementation (i.e. percentage of full CMMI ML2 compliance using the staged representation) goals for each month as a cumulative quantity, while the line graph shows the actual compliance achieved.

Figure 1 shows the early gains made from prototyping some of the processes and tools in the SEPG and then piloting tailored versions to the projects. It also shows a slight dip in August as the organization played catch-up on their QA audits (by outsourcing); otherwise, the PI effort was executed according to the plan.

While it was not technically and fully compliant until September, it had achieved a high degree of institutionalization well in advance since the majority of many of the PA's had already been up and running for quite some time (the first process was rolled out in February). This is a major benefit derived from implementing an incremental and iterative approach.

The compliance profile from the PA perspective is shown in Table 3. By June (four months before the actual appraisal), a large percentage of each PA had been implemented. The percentages were based not only on whether a particular process was being performed but on the coverage of work products completed as well. For example, complete credit was not claimed for REQM SP 1.4 until the RTM was completed. The appraiser's perspective is similar but not as rigid. Appraisers want to ensure that processes and work products meet the intent of the model, and they want to see *sufficient* evidence that the processes are being followed. So, for example, an RTM in the process of development with substantial progress made is acceptable and practical. The reason we chose a different interpretation was to drive the projects toward completing their work products. As a result, one project was able to complete its RTM by the appraisal, while the others had made significant progress toward completing theirs. In the end, all were able to claim full credit for their RTM.

From the trends in Table 3, one might expect weaknesses in PPQA and CM since they lagged the other PA's in reaching comparable compliance levels. They were among the last three processes to be rolled out. The appraisal did note a weakness in PPQA, but none in CM.

Although the original schedule called for a February 2007 appraisal, the Lead Appraiser felt that MORI had already

SM PSP and TSP are service marks of Carnegie Mellon University.

achieved a high state of readiness much sooner. MORI achieved ML2 (staged representation) on October 4, 2006, in nine months with six global and several PA's strengths with only two weaknesses. This result further reflects how a commitment to quality and continuous improvement combined with a more agile approach can help you reach your improvement goals in dramatic fashion.

Summary

As a result of this PI effort, MORI learned many lessons that spanned the entire PI life cycle. Creating a streamlined PI effort is definitely possible when you follow a more agile approach. Implementing an incremental/iterative approach, piloting prototypes to the organization early and often, leveraging industry standards and examples, and identifying and using metrics to monitor and adjust the plan and schedule as needed are all ways to develop processes in a highly responsive manner. Reducing the overall impact to the organization is possible when you outsource process development, implement well defined processes, and provide the right mix of training, mentoring, and bootstrap services. Using an agile approach can yield significant and even unexpected results over more traditional methods.◆

References

1. Rico, David F. "How to Estimate ROI for Inspections, PSP^SM, TSP^SM, SW-CMM, ISO 9000, CMMI." <u>Software Tech News</u> 5.4 (2002) <www.softwaretechnews.com/stn5-4/inspections.html>.
2. Rico, David F. "Capability Maturity Model Integration – CMMI Cost Model." <u>David Rico</u> 2001 <http://davidfrico.com/cmmi.pdf>.
3. IEEE. <u>IEEE Software Engineering Collection</u>. IEEE, 2003 <www.ieee.org/>.
4. Humphrey, Watts S. <u>Managing the Software Process</u>. Addison Wesley, 1990.
5. SCAMPI Upgrade Team. <u>Standard CMMI Appraisal Method for Process Improvement A, Version 1.2: Method Definition Document</u>. SEI, Carnegie Mellon University, 2006.
6. Juran, J.M. <u>Juran on Quality by Design</u>. Free Press, 1992.

About the Authors

Nelson Perez is president of Sierra's Edge, Inc., and was the architect and author of all the policies and process assets at MORI Associates. He has more than 20 years of experience and has worked the entire life cycle of and held numerous management and engineering positions on such high-visibility programs as the B2 Stealth Bomber, National Aeronautics and Space Administration Space Shuttle, and Homeland Defense. Perez has co-authored one patent, and his first PI effort helped garner the U.S. Air Force/TRW (now part of Northrop Grumman) Special Operation Force Extendable Integration Support Environment program the U.S. Air Force 21st Partnership Century Team Quality Award.

Sierra's Edge, Inc.
P.O. Box 3176
Silver Spring, MD 20918-3176
Phone: (301) 801-0740
E-mail: nelson@sierrasedge.com

Ernest C. Ambrose is the SEPG lead as well as a project lead for MORI Associates, Inc., a woman-owned, Small Business Administration 8(a) certified, small disadvantaged business providing professional consulting services for sophisticated information technology solutions, program support, staff augmentation, and engineering operations for government agencies and Fortune 500 companies. MORI is the recent recipient of Maryland's Top 100 Minority Business Enterprise Award and is ranked as one of Washington Technology's FAST 50 companies.

MORI Associates, Inc.
6701 Democracy BLVD
STE 206
Bethesda, MD 20817
Phone: (301) 493-6674
E-mail: eambrose@
moriassociates.com

Controlling Organizational Change: Beyond the Nightmare

Deb Jacobs
Focal Point Associates

The only thing constant is change! Organizational change can be a nightmare; this is especially true with process improvement. There are many challenges connected with transitioning new ideas and changes within an organization. There are numerous tried, true, and innovative theories for achieving organizational change. By applying a combination of these theories or considering the implications of each, it can help make organizations flexible enough to be prepared for and realize change as it happens.

As Ann popped another piece of candy in her mouth, she steeled herself in preparation for starting the weekly Engineering Process Group (EPG) meeting. Lately, the meetings were becoming pretty miserable. Instead of progress reports, the meeting tended to be a forum for complaints. Usually, complaints were very welcome since they alerted the group to problems that needed to be resolved, but the complaints were becoming less constructive and more destructive. She had done all the things the experts suggested to make the EPG meeting and the process improvement effort more successful. She gave everyone a chance to participate, listened to complaints and tried to resolve them or elevate them as needed, worked closely with projects, and made sure there was generous participation in the meetings from all projects. Ann generally put 250 percent into making the process improvement effort work for the organization – she even provided goodies for the meetings. None of these things seemed to be working, and she was at a loss as to where to turn.

Coming back from her mind wandering, Ann decided she better just get started with the meeting. She put her best face forward, smiled, and said, "Good morning everyone. Let's get the meeting started." She heard a smattering of good mornings from around the room.

"I think the only one missing is John. He called me earlier to say he had an emergency on his project and couldn't make it," Ann continued. "Let's start the

> ## "Controlling the changes that occur during a PI effort is one of the most difficult but one of the most important aspects of the effort."

meeting out talking about what progress has been made on the actions assigned to each action team…"

Before she could finish, Mike blurted out, "Ann, we haven't been able to make much progress at all. With all the problems we've been seeing throughout the organization, we seem to be taking three steps forward only to be pushed five steps back again. If we don't resolve the underlying issues, we're wasting our time." Mike was a project manager from one of the company's highest profile projects. He always had great ideas and worked well with the EPG. He was very enthusiastic about the effort since he saw how much it would help his project be more successful. However, like the rest of the group, Mike was becoming increasingly frustrated.

The room felt like it was closing in on Ann and she didn't know what to do, recovering quickly since this was not the first time she had been up against the wall, Ann asked, "So give me some ideas. I agree with you Mike but we need another approach."

She heard the general shuffling, papers rustling, and covered coughs around the room as she paused for ideas. Ann decided it was time to lay it on the line. "Folks, this may be our last chance. I heard through the grapevine that Mr. Jones is talking about cutting our funding for the process improvement effort. He hasn't seen much progress lately and has heard about the problems we've been encountering making the changes needed in this organization to implement the processes. At first it seemed as if we were going gangbusters, then we ran out of gas without a gas can."

Sound familiar? It is for many organizations trying to make much needed organizational changes. Too many organizations experience unorganized, uncontrolled chaos during process improvement efforts. So what is the answer to resolving these typical problems [1]?

The Nightmare of Organizational Change

Controlling the changes that occur during a process improvement effort is one of the most difficult but one of the most important aspects of the effort. There are numerous theories that can be used in combination to assist organizations in the organizational change effort that are key to the entire process improvement effort.

Rational Change

Peter F. Drucker, called the father of modern management, stated the following:

> It is not true, as a good many industrial psychologists assert, that human nature resists change. On the contrary, no being in heaven or earth is greedier for new

Table 1: *Handwriting on the Wall*

Change Happens
Someone keeps moving the cheese.
Anticipate Change
Get ready for the cheese to move.
Monitor Change
Smell the cheese often so you know when it is getting old.
Adapt to Change Quickly
The quicker you let go of old cheese, the sooner you can enjoy new cheese.
Change
Move with the cheese.
Enjoy Change
Savor the adventure and enjoy the taste of new cheese.
Be Ready to Enjoy Change Quickly, and Enjoy it Again
Someone will keep moving the cheese.

things. But there are conditions for man's readiness for change. The change must appear rational to him ... [2]

The key is that any change must appear rational especially when trying to effect an entire organization by making it more effective with documented and used processes. You must appeal to a staff's rational side. It is a matter of finding the right methods to use to make staff realize that the changes are rational and work in their favor.

At the onset, process engineers must realize that there will be resistance to change but if it is managed and promoted properly, the resistance can be minimized and controlled.

Resistance to Change

There are many methods of identifying typical resistance behaviors in order to manage and minimize resistance to changes introduced as part of the process improvement effort. Spenser Johnson, M.D. in his book, *Who Moved My Cheese?* said, "Movement in a new direction helps you find new cheese [3]." This is especially appropriate for process maturity. Process maturity is constant change and evolution. Sometimes change is in a totally new direction or it can be in the same positive direction depending on where an organization is in the process improvement lifecycle. With any change comes adjustment in varying degrees to the way things are done; in other words, finding new cheese. *Who Moved My Cheese?* describes the reactions of four mice, Scurry, Sniff, Haw, and Hem when change occurs in their lives, symbolized by moving cheese in the maze. The cheese represents elements in life such as career, happiness, financial success, relationships, peace of mind, health, etc. Table 1 illustrates Johnson's handwriting on the wall. This is a simplified version of various human behaviors reacting to change, but if these basic human elements are taken into consideration when managing and controlling change, the degree of success goes up substantially.

Another popular theory in studying human behavior and resistance to change is the Everett Roger's Adoption Curve [4]. Roger's theory concerns how new ideas are disseminated and accepted by groups of people. Even though many have shown that there are some issues with the Adoption Curve, it is still a good rule of thumb for adoption of new technologies and ideas. Simply realizing that change is adopted at different rates by
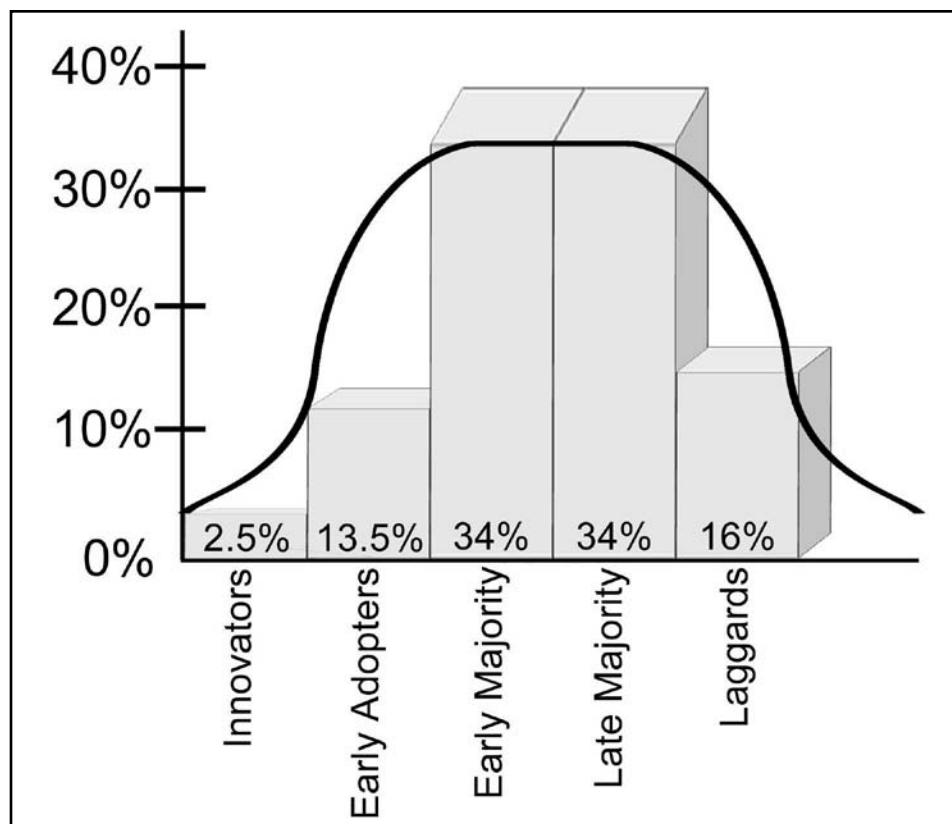
Figure 1: *Roger's Adoption Curve*

various people can help with planning and control of the organizational change effort. Roger's theory holds that given a normal population distribution, people accept new ideas and innovation at a different rate. He defines an innovation as *an idea, practice, or object that is perceived as new by an individual or other unit of adoption* [4]. He defined various adopters as follows:

- 2.5 percent ==> innovators.
- 13.5 percent ==> early adopters.
- 34 percent ==> early majority (early mainstream).
- 34 percent ==> late majority (late mainstream).
- 16 percent ==> laggards.

This distribution of adopters results in a bell curve, as shown in Figure 1.

The innovators are the few who first take up a new practice or listen to a new idea. The early adopters come along next and are great for communicating the effort to others since many times these are the social leaders. Once the social leaders take up a new idea, the early majority takes up the idea fairly quickly followed by the late majority. Finally, the laggards are typically the last to consider a new idea; they tend to adopt innovation very slowly. You must always account for a few laggards to bring up the rear.

A *chasm* has been defined between the early adopters and the early majority.

Figure 2 illustrates the *chasm*. This *chasm* implies that the transfer of information flows from innovators to early adopters easily but that it is difficult to translate

Figure 2: *Roger's Adoption Curve Chasm*
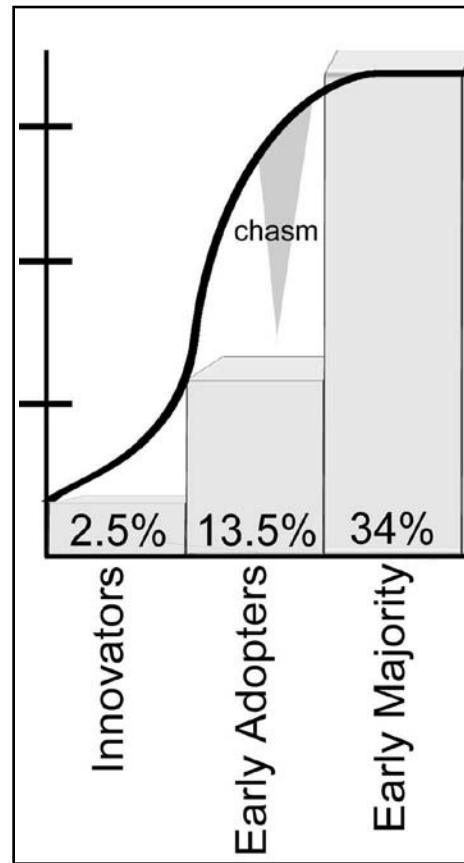
Adoption Curve Category	Learning Curve	Effort Required for Adoption	Continued Support of Effort	Retention of Adopter
innovator	rapid learners	no recruiting effort	may be low	steadfast
early adopters	rapid learning	minor effort	moderate; in spurts	dependable
early majority	reasonable learning curve	substantial effort	higher and continuous	fickle
late majority	trainable but slow	major effort	highest continuous support	brittle
laggards	typically uninterested	typically uninterested	not feasible	

Table 2: *Adoption Curve Effort and Learning*

that into action and acceptance by the early majority which is sometimes called the early mainstream. This is where the marketing of the effort and the resulting processes becomes most important; you have to get the word out to get results.

Universities use this curve to define the effort required for each category of human behavior to recruit and retain adopters of new information and provide the appropriate training. Table 2 summarizes some of the data collected and used by universities.

The Convergence Model, Roger's latest model, is also worth exploring. It emphasizes the need for a continual process of interpretation and response, leading to an increased degree of mutual understanding between sender and receiver [4].

Roger's human behaviors are not too much different than the characters in *Who Moved My Cheese?* Scurry is the innovator who goes into action immediately upon change or something new – he finds the new cheese and moves things along. Sniff is the early adopter. He sniffs out changes and systematically searches out new cheese (don't let the fact that he's a mouse fool you). Haw is the early-to-late majority who is reluctant to change and fearful but overcomes fears and moves with the change – he's the one that puts the writing on the wall hoping Hem will follow him. Finally, Hem is the laggard who says no change under any circumstances – his arms are folded against any change.

The key to using these two theories is to know where to focus your resources. It also tells you who may be interested in helping advertise your effort; word of mouth and advertisement of even the smallest changes are key ways of getting others on board for organizational change. Be careful not to pigeon-hole anyone though because the surprise may come when a perceived innovator becomes a laggard or jumps ship when the honeymoon period is over. Even more surprising is when a perceived laggard becomes an early adopter, thus becoming one of your best assets to promoting change and overcoming resistance.

The Science of Organizational Change

Experts have been relating organizational change to other arenas. Organizational change can be better managed by studying other unlikely, more scientific arenas such as Chaos Theory, open systems as related to biology and planets, quantum physics and leadership, and anthropology. These theories not only help us further understand methods for becoming successful organizations but help make us flexible enough to make effective changes.

The Chaos Theory

The Chaos Theory describes systems apparently disordered but having an underlying order. The theory is about finding the underlying order in apparently random data. In ancient Greece thousands of years ago, the cause and effect rules were introduced as a philosophical belief. Sometime around the 1500s, this concept was accepted as a scientific theory. Isaac Newton's laws implied that everything that would occur would be based entirely on what happened right before. Henry Adams is credited with first describing chaos as, "Chaos often breeds life, when order breeds habit." He also said, "Chaos is the law of nature;

Order is the dream of man."

In 1846, the planet Neptune was discovered which had been predicted from the observation of deviations in Uranus' orbit. Oscar II, then the King of Sweden and Norway, initiated a mathematical competition in 1887 to celebrate his 60th birthday in 1889. He challenged anyone to prove or disprove that the solar system was stable. Henri Poincaré, sometimes called the *Father of Chaos*, was awarded the prize for his three-body problem in celestial mechanics where he provided the first mathematical description of chaotic motion. However, when a colleague found an error in his theory, the prize was taken away until he could find a new solution. After much consultation with colleagues, he found that there was no solution including use of Newton's Laws. Poincaré had been trying to find order in a system where there was no order; this error is now regarded as marking the birth of Chaos Theory.

Edward Lorenz, a meteorologist at Massachusetts Institute of Technology, has been called the first true experimenter in chaos in the 1960s because of his work on a weather prediction problem. Lorenz set up a computer with 12 equations to simulate the weather. This program theoretically predicted the weather, but in 1961 when he wanted to see the sequence again but wanted to save time, he started in the middle of the sequence and let it run. This sequence diverged from the original pattern. What he found was that the number had been stored with six decimal places in the original sequence and when he re-ran the program, he rounded the six decimals to only three decimal places to save paper. Where he should have gotten a sequence very close to the original, it had a huge effect on the resulting pattern. This is known as sensitive dependence on the initial conditions, which Lorenz found changes the long-term behavior of a system. He found that small changes on things lead to changes on a large scale. It was the classic example of chaos.

So, what does that have to do with managing organizational change? You cannot always predict what a system will do next, but you can put things in motion in smaller ways to perpetuate changes on a large scale. This premise can be used effectively during process improvement effort planning and throughout the effort to start the project off well. It can also be used effectively during continuous improvement to monitor how each change impacts the organization.

How a pattern eventually looks is

dependent upon the precision of the predicted initial conditions. Small inaccuracies or changes can have huge effects. During planning, a key element is setting the initial conditions to perpetuate change in the organization. By considering the importance of setting initial conditions such as management expectations, staff expectations, selection of actions to tackle, and who should be involved and to what degree in the effort, an organization will be able to provide a much more accurate action plan. During continuous improvement efforts, there will be many lessons learned to draw from when considering the initial conditions for change.

In his book *Chaos: Making a New Science*, James Gleick related chaos to the motion of a water wheel and found the following:

> …water drips steadily into containers hanging on the wheel's rim. Each container drips steadily from a small hole. If the stream of water is slow, the top containers never fill fast enough to overcome friction, but if the stream is faster, the weight starts to turn the wheel. The rotation might become continuous. [5]

This holds true with process improvement as well. You must get things moving fast enough and keep the momentum going in order to make the organizational changes needed.

Gleick also points out the following:

> … if the stream is so fast that the heavy containers swing all the way around the bottom and up the other side, the wheel might then slow, stop, and reverse its rotation, turning first one way and then the other. [5]

In other words, the spin becomes chaotic. As buckets pass under the water, how much the buckets fill depends upon the speed at which the wheel is turning.

The key is to keep the momentum going fast enough to effect change but not so fast that things become chaotic. On the other hand, with process improvement efforts, you can never let things settle down enough to stop the momentum since sometimes it is even harder to get the momentum going again – or it can reverse the progress already made. During subsequent planning and monitoring of a process improvement effort, this theory can be useful to ensure

that the momentum of the process improvement is continuous and progressive.

Open Systems

Many are starting to look at organizations as open systems to help define their structure. Dr. Helene Uhlfelder holds that organizations are much like the human body and our planet. In other words, examples of open systems are the human body and or solar system where the human body is composed of interacting biological cells and our solar system with planets, stars, etc. They are like an organization where each is engaged in active transactions with their environment. Uhlfelder states that, "An open system has certain characteristics that need to be understood if one is going to work in and change it" [6]. Some of the characteristics that she defines include the following:

1. Open systems are porous and have permeable boundaries.
2. Open systems are interdependent with surrounding systems and are composed of interdependent parts.
3. Open systems need to be dynamic and fluid to survive.
4. Open systems are interactive with their environment and must be adaptable.
5. In an open system, the whole is greater than the parts.

A key to organizational change in

Uhlfelder's theory holds that closed systems can result in entropy:

> Entropy is an inverse measure of a system's capacity for change and means the system will eventually die from a lack of energy. Because open systems can import energy and can be dynamic and fluid, open systems can grow and change. [6]

Uhlfelder contends that systems should never strive for equilibrium and should always maintain a certain amount of dis-equilibrium as a necessity to foster change. As with ecology, she finds that departmental interdependency is a key to success; in other words, "what happens in one department affects what happens in another department. The key is for an organization to become an open system that is adaptive to allow "input and output with the environment, political and social institutions, and world events" [6]. Uhlfelder's Open System Organizations is illustrated in Figure 3.

For process improvement, each of the aspects of an open system should be considered during planning and monitoring of the process improvement effort. The impact of each of these key elements can have a significant impact on the effort. A process improvement effort should be handled just like any other project, hence, a risk management strategy

Figure 3: *Uhlfelder's Open System Organizations*
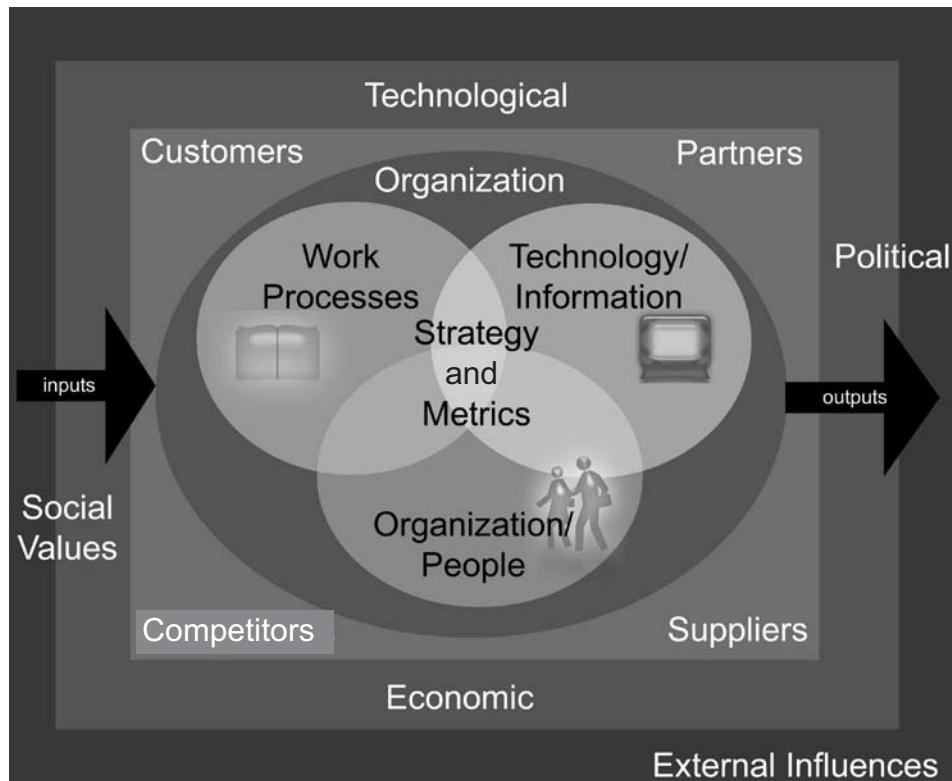
should be developed. Each of the elements in an open system should be considered when identifying risks, and the impacts to each element should be considered when developing a risk mitigation plan.

Each organization is very different so the key areas to consider will vary based on the organization's goals and objectives. The process improvement action plan is the best place to develop a strategy for handling each of the elements that are key to that organization.

Quantum Physics and Leadership

Margaret Wheatley in *Leadership and The New Science* applies quantum physics to leadership. The new science of quantum physics describes a universe *where order and change, autonomy and control were not great opposites that we had thought them to be. It was a world where change and constant creation signaled new ways of maintaining order and structure* [7].

Wheatley relates the *field theory to organizations which asserts that fields are unseen structures, occupying space and becoming known to us through their effects*. She holds that, *All employees, in any part of the company, who bumped against the field, would be influenced by it … their energy would link with the field's form to create behavior congruent with the organization's goals*. If we *think of ideas as fields*, it will permeate the entire organization. She states that, *We need all of us out there, stating, clarifying, discussing, modeling, filling all of the space with the messages we care about. If we do that, fields develop – and with them the wondrous capacity to bring energy into form* [7].

This is especially appropriate for process improvement where participa-

tion of the process users is a critical key to success of each process and the process improvement effort as a whole. Wheatley says that an organization's vision is actually a culmination of all the people who make up an organization as opposed to being handed down from management. The concept of ownership is a key where she discusses that *the best way to build ownership is to give over the creation process to those who will be charged with its implementation*. She holds that *It doesn't work to just ask people to sign on when they haven't been involved in the design process, when they haven't experienced the plan as a living, breathing thing* [7]. Participation in creating processes is a must for process improvement; it is a must in understanding and accepting a process.

Anthropology

Anthropology is being seen as more and more important in the world of business. Primarily, anthropology is the study of

> **"Organizations must be prepared to change quickly in response to changing markets and must be flexible enough to introduce new skills and technologies as appropriate."**

human behavior. There are many subfields in anthropology; the relevant one to organizational change is business anthropology. Business anthropology is the study of human behavior in complex organizational structures. Many universities are offering graduate degrees in business anthropology. Additionally, many large companies and the government are employing business anthropologists such as IBM, Intel, Microsoft, General Motors, and Xerox to name a few. They study the behavior of both the client and the company staff to determine what typical shared behaviors are demonstrated. These shared behaviors help in determining a strategy for effecting organizational change. By understanding what makes a certain group of people motivated and what expectations they have, you can determine the best methods for

employing processes and making them work within that particular organization.

Throughout history, each age demonstrated a shared behavioral pattern that defined the essentials in all realms of life (home, work, and recreation). Survival in each age was dependent upon differing factors. In the Stone Age, the keys to survival were food and shelter, where in the Industrial Age they were factories and equipment. In today's Information Age, survival means having the appropriate technical skills and knowledge as well as learning to manage shared behavioral patterns in order to make changes in a world that is changing at Internet speed. Organizations must be prepared to change quickly in response to changing markets and must be flexible enough to introduce new skills and technologies as appropriate. When developing processes for an organization, these premises should be considered to ensure that processes are as flexible as needed based on the environment and changing world.

The Perspective Factor

MC Escher is famous for his optical illusion art from the 1940s to the 1970s. He was well known for his impossible structures. A favorite is called Relativity that basically tells us that what you see is relative to where you're standing. When dealing with others' realities, we have to see things from their perspective and make it relative to them.

Each stakeholder is going to see something different; it is important to discuss a process based on their perspective. It all depends upon each of our perspectives. How you see something depends on what vantage point you are coming from. We all look at things differently based on our background, education, experience, and simply from where we are standing at the moment. It is key to process improvement, that process development staff and users work as a team to develop effective processes in order to develop effective products. If we look at things from each other's vantage point, the chance of success grows by leaps and bounds. Open communications and respect for each other's position is crucial for success.

As a process engineer developing processes for a project or organization, the key is to let the user know that when you're done you can simply walk away, but they need to be able to use the developed processes to accomplish their job. The process engineer must work with users to make the processes work for them, at the same time keeping in mind their perspec-

Table 3: *Marketing Trends*

Rank	Technique
1	Affiliate programs
2	E-mail to customers
3	Public relations
4	Television
5	Outdoor
6	E-mail (opt-in lists)
7	Magazines
8	Radio
9	Direct mail
10	Sponsorships
11	Buttons
12	Banners
13	Newspapers

tive as well as that of the selected model or methodologies requirements.

Marketing Processes and Process Improvement

By studying marketing trends and methods, we can use some of the more effective methods for marketing process improvement and the resulting processes. Table 3 ranks some marketing methods according to Forrester Research [8]. Their ranking is based on Forrester's analysis of how popular or how often a method is used, and how effective for the organization that method is when marketing their products. The key for process improvement is knowing what methods are available and their likelihood of being effective in a specific organization. Many variables need to be considered such as size, office distribution, communication methods available, cost of each method, and overall impact of each method to the organization.

Many of these can be very effective in marketing process improvement throughout an organization. There are many tools and strategies that can help with the marketing aspect of processes improvement, hence, easing organizational change. Appropriate dissemination of information is key to the success of process improvement. Table 4 lists several successful methods.

Summary

Organizational change can be difficult. It is a matter of finding the right methods to use to make staff realize that the changes are rational, their perspective has been considered, and the changes work in their favor. To paraphrase Drucker, human nature does not resist change if it seems rational. We are, in fact, by nature ready to try new things. Looking at some seemingly diverse arenas, such as the ones addressed in this article, helps us control change. The theories and methods discussed help us understand what elements need to be considered when planning and monitoring a process improvement effort. All aspects of an organization must be considered when making changes.

These theories can be combined effectively to assist organizations in the organizational change effort that is key to the entire process improvement effort. The characters Ann and Mike are composites of people typical to many organizations; they want to work in a rational environment that is organized and exhibits controlled chaos rather than

Portals	Single integration point to disseminate information – knowledge board.
Business Intelligence	Virtually provides information on demand – access to a variety of data sources.
Focused Newsletters	Great way to disperse information – training and knowledge dissemination.
Bulletins	Encourage participation, announce events and successes, and disseminate information.
Posters	Advertise the effort, get exposure, keep everyone informed.
E-mails	Great for information dissemination – do not overuse or abuse – keep it light.
Staff Meeting Announcements	Make presentations, initiate efforts, show status, and advertise successes.
Polls	Determine if changes are accepted and disseminated.
Affiliation Formation	Social leaders, project managers, movers and shakers on projects or groups.
Keeping an Eye on the Future	If you do not keep a close eye on the flurry of activity in the technology market, you will get left behind as technology progresses.

Table 4: *Tools and Strategies for Effecting Organizational Change*

uncontrolled chaos. Chaos is everywhere, but we can control that chaos, and it can be very effective as opposed to nightmarish.◆

References

1. Jacobs, Deb. <u>Accelerating Process Improvement Using Agile Techniques</u>. Auerbach Publications, 2006.
2. Drucker, Peter F. <u>The Essential Drucker: The Best of Sixty Years of Peter Drucker's Essential Writings on Management</u>. Harper Collins, 2003.
3. Johnson, Spencer. <u>Who Moved My Cheese? An Amazing Way to Deal With Change in Your Work and in Your Life</u>. Putnam Publishing Group, 1998.
4. Rogers, Everett. <u>Diffusion of Innovations</u>. 5th ed: Free Press, 2003.
5. Gleick, James. <u>Chaos: Making a New Science</u>. Penguin, 1998.
6. Miller, Lawrence M., and Helene F. Uhlfelder. <u>Change Management: Creating the Dynamic Organization Through Whole System Architecture</u>. Miller Consulting Group, 1997.
7. Wheatley, Margaret J. <u>Leadership and the New Science: Discovering Order in a Chaotic World</u>. 3rd ed. Berrett-Koehler Publishers, 2006.
8. <u>Forrester</u> <www.forrester.com>.

About the Author

Deb Jacobs has more than 30 years of experience in information technology including system/software engineering, project management, process improvement, and proposal development with a bachelor's degree in computer science. She has helped make many organizations more successful in development and management. Notable successes include leading a successful Capability Maturity Model (CMM) Level 3 effort in one year, successfully reengineering struggling projects, mentoring new managers, developing numerous technical papers, and gaining new business for companies through winning proposal development. Jacobs is the author of several published technical articles as well as the popular Process Improvement book *Accelerating Process Improvement Using Agile Techniques*. She has been a member of the Software Engineering Institute (SEI) CMM upgrade, CMM Integration℠ (CMMI). She is the former SPINOUT newsletter editor/originator, a former CERT® (Combined Environmental Reliability Test) Conference Chairperson, infotec Deputy Software Tracks Chair, SEI CMMI contributor, and is a member of the CROSSTALK Editorial Board

Focal Point Associates
14301 First National PKWY
STE 411
Omaha, NE 68154
Phone: (402) 445-4700
E-mail: djacobsfpa@aol.com

® CERT is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
℠ CMM Integration is a service mark of Carnegie Mellon University.

Net-Centric Conversations:
The Enterprise Unit of Work

Harvey Reed and COL Fred Stein (Ret.)
MITRE

Net-centric warfare and Net-Centric Operations (NCO) require systems and systems of systems to be increasingly agile. The challenge is how to design and modify such agile systems. This article suggests that the concept of Net-Centric Conversations (NCC) can be used to increase and track the agility of systems, in support of both humans and machines. An NCC is an operational mission thread or business process expressed, according to a set of five organizing principles. These principles support a formal description of the people, machines, weapons, and sensors, and their cooperative relationship that produces a net-centric capability. The term conversation *is deliberately used to capture the essence of an NCC's ability to describe the interactions between the participants, both humans and machines, involved in the mission thread or business process. Each NCC is version controlled, ensuring that all participants are aware of relevant changes in others, allowing continuous configuration management and build-up of trust between them.*

This article discusses a practical and innovative means to create and manage net-centric capabilities that span systems and people. The intended audience includes both acquisition and operational leaders. The concepts are most germane to those who have tried to field Web services or publish/subscribe services and who have tried to integrate these services with other likewise programs. There is almost a universal frustration with the current program-centric approach to constructing net-centric capabilities that span systems and people; this article proposes an alternative.

Net-centric warfare and NCO are based on the existence of a highly connected force capable of leveraging the interdependent relationships among sensors, shooters, and decision makers, all enabled by information technology. Net-centric capabilities (such as new *kill* or *supply chains*[1]) are generated by relating multiple weapons, sensors, and people together, either permanently or temporarily. These net-centric capabilities require supporting complex relationships. Traditional bilateral interface exchanges such as Interface Exchange Requirements and Interface Control Documents (ICD) are insufficient to describe and manage complex net-centric capabilities. Since the Department of Defense (DoD) has no formal mechanism to describe and manage such relationships, it is difficult to maintain trust among participants, which slows adoption of NCO.

To construct the kind of net-centric capabilities discussed, the primary focus is on creating and managing relationships among the participants (sensors, shooters, decision makers, supporting machines, and people) who use the network to exchange messages. We are not talking about the network itself (routers, bridges, pipes, etc.), except for the intersection cases where a firewall or proxy would interfere with the exchange of messages. This distinction is important because the network itself

> *"In a multilateral NCC, many program offices will contribute services, and the impact analysis must alert all potentially affected program offices of a pending change that the group needs to discuss."*

should be largely unconstrained in how it delivers services. We will assume the network service is highly available and has the ability to deliver messages.

Further, individual systems with Web and messaging technologies such as enterprise service bus (ESB) do not construct net-centric capabilities in and of themselves. They provide connection mechanisms; the hard part is describing, recording, and managing the relationships you want to create.

Figure 1 highlights the difficulty of constructing a net-centric capability on top of a network using bilateral techniques to construct and manage relationships among participants. Three stovepipe systems are individually exposing services (such as Web services, or publish/subscribe services) to exchange messages. The messages can contain targeting information from a Command and Control system or perhaps logistics, as well as personnel information from an operations support system. The messages can be exchanged with a variety of techniques including publish/subscribe, and/or Web services.

The desired end-state capability spans all the participants, including the services and end-users. Current interface agreements are bilateral, forcing us to use at least three agreements in this simple example. Further, there will be multiple negotiations with security officers for each of the systems since the messages exchanged will likely be passing through firewalls, proxies, etc.

Once the capability is constructed, it is difficult, if not impossible, to change because there is no focal point for change control or impact analysis. Adding to this, there are too many separate agreements to change in a coordinated fashion.

The alternative to creating and managing separate bilateral agreements is to create a single multilateral agreement, as shown in Figure 2. This centralizes change impact analysis and aids in coordinating change. Change is necessary to implement agility to respond to unanticipated events. It is only with a single multilateral agreement and correspond-

ing agility metrics that agility can be measured and improved over time.

Publish-01 is the name of the net-centric conversation that binds all the participants listed in the legend.

NCC: The Definition

An NCC is a persistent, multi-party agreement describing the relationships between sensors, shooters, decision makers, and other participants that create a net-centric capability. This agreement is ideally stored and managed in an NCC discovery service as an extension to the metadata environment for the enterprise. NCCs are supported by the following five organizing principles.

NCC Principle 1

An NCC is described, registered, and discoverable. NCC is a binding layer (see Figure 3) for the messages and mission services (warfighter and business capability) that, in turn, use enterprise services. The NCC describes critical roles for people, support doctrines, and procedures; it is entered into an NCC registry so that impact analyses can examine any proposed changes. This impact analysis must support both low-level and high-level changes. In a multilateral NCC, many program offices will contribute services, and the impact analysis must alert all potentially affected program offices of a pending change that the group needs to discuss.

Stability of message structures are critical to the stability of NCCs. Individual mission services can change their implementation with no impact on participants; however, the same cannot be said of message structure. Future message structures will be defined as extended mark-up language (XML) schemas whose vocabulary will be well defined and explained by community of interest (COI) data panels. The value of XML is that it is flexible and can allow extensions, but changing the core structure or the vocabulary itself will have far-reaching effects.

NCC Principle 2

An NCC is described by message exchanges of participants. An NCC groups machines and users in a transitive, multilateral agreement to produce a net-centric capability. As shown in the NCO examples and in the notional scenario in Figure 4 (see page 20), the NCC is defined by the exchange of messages; this information is what is recorded in the NCC's entry in the NCC registry, as well as supporting Concept of Operations (CONOPS) and other ancillary materials to aid in the understanding and measure-

Currently, stovepipe systems make agreements with their nearest neighbors – but no mechanism to describe and manage an entire thread of processing.

LEGEND
S1 – Web service
S2 – Web service
S3 – Web service
M1 – message
M2 – message
M3 – message
U1 – user
P1 – security perimeter 1
P2 – security perimeter 2
P3 – security perimeter 3
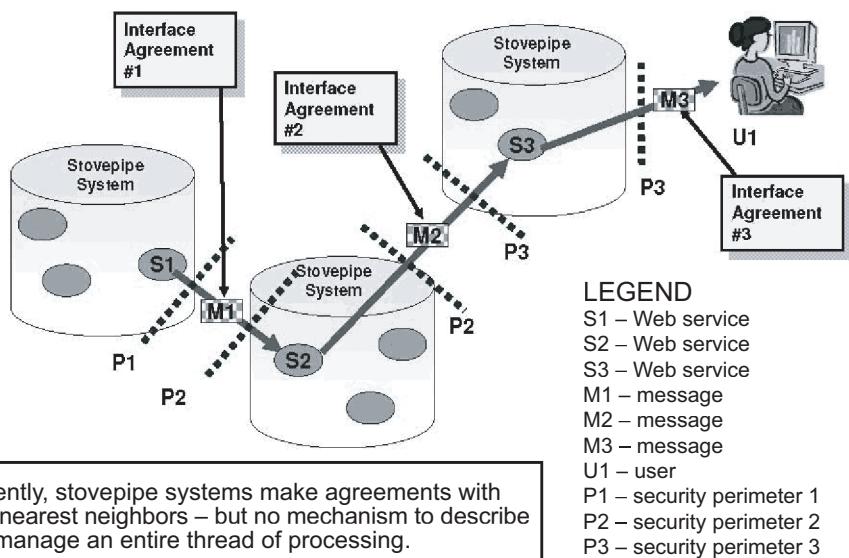
Figure 1: *Stovepipe Systems – Separate Agreements*

One NCC "Publish-01"

NCCs provide a means to create one agreement that covers many participants. This assures we can manage change over time.

LEGEND
S1 – Web service
S2 – Web service
S3 – Web service
M1 – message
M2 – message
M3 – message
U1 – user
P1 – security perimeter
P2 – security perimeter
P3 – security perimeter
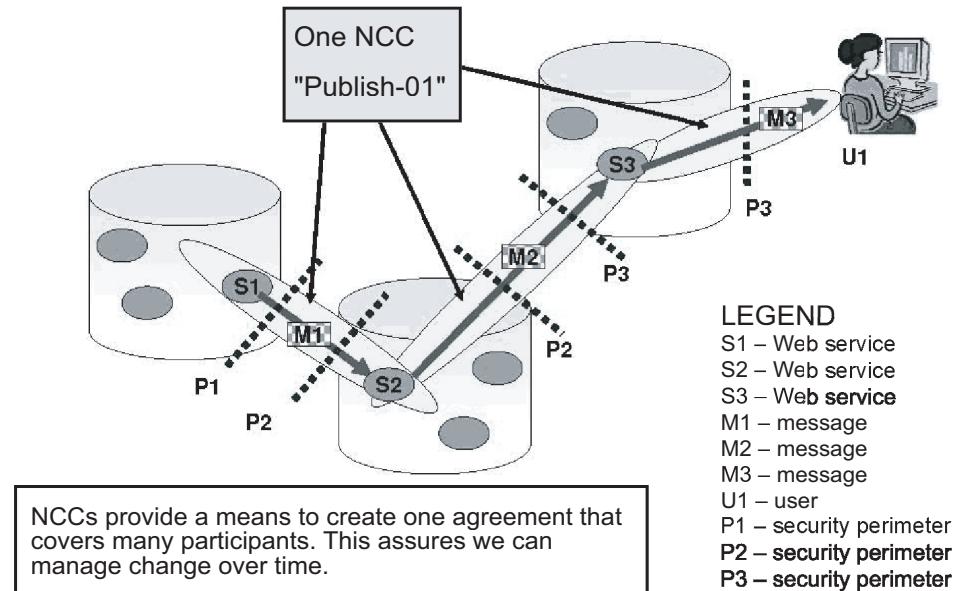
Figure 2: *NCCs – Single Agreement*

An NCC is described, registered, and discoverable, and represents the persistent net-centric capability.

Impact Analysis (both directions)

NCC
(buisness process, chat, publish/subscribe, etc.)

Mission Information
(messages)

Mission Services
(including data and sensors)

Enterprise Services
(messaging, storage, etc.)

NCCs are registered and *point to* dependent services and metadata.

When a service or metadata is versioned, we can do simple impact analysis.
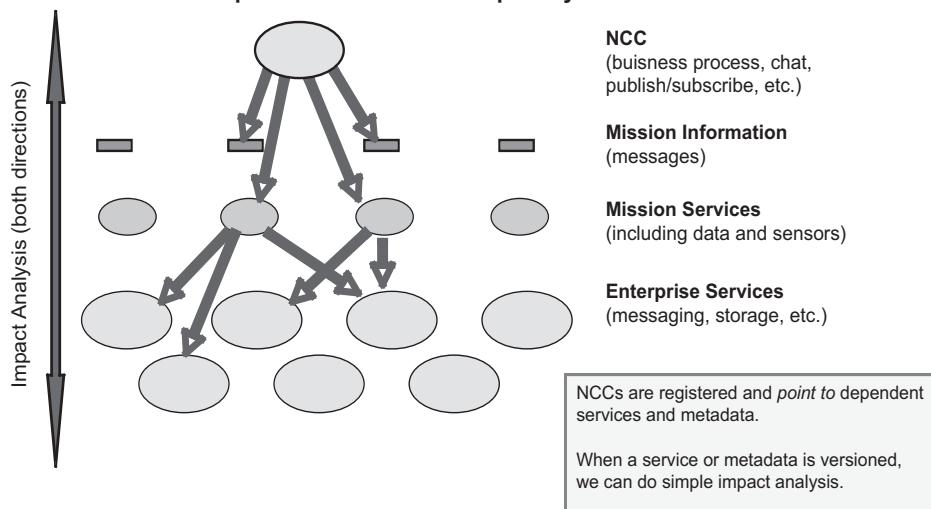
Figure 3: *NCC Principle 1*

An NCC has both human and machines as participants and is completely described by the set of possible message exchanges between them.
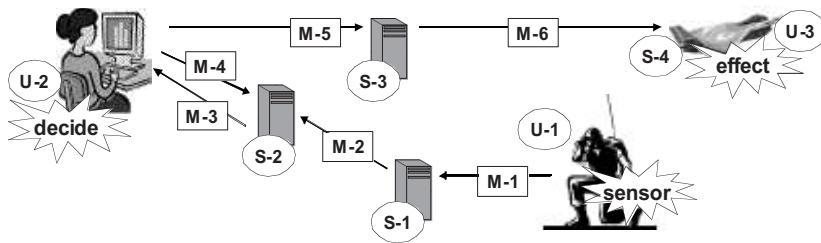
NCC Name	Users	Systems	Messages	CONOPS and Doctrine	Mission Effectiveness KPIs
Laser-target-01	U-1, U-2, U-3	S-1, S-2, S-3, S-4	M-1, M-2, M-3, M-4, M-5, M-6	SOF	1. Ave time to target 2. Ave accuracy

Figure 4: *NCC Principle 2*

An NCC has an agility profile derived from the agility metrics of the participants and messages.

(NOTIONAL)

| Participant Type	Participant Name	Minimum time to develop/train	Organization	Minimum time to reconfigure
User	U-1	Train=10 days	AETC	n/a
Service	S-1	42 days	Contractor-1	n/a
Service	S-2	67 days	Contractor-2	13 days
Service	S-3	83 days	Contractor-2	5 days
Message	M-1	120 days	COI-1	n/a
Message	M-2	160 days	COI-1	n/a
Message	M-3	200 days	COI-2	n/a
Perimeter	P-1	n/a	AFNOSC	Firewall=90 days
Perimeter	P-2	n/a	DISA	Firewall=60 days
Perimeter	P-3	n/a	DISA	Firewall=45 days

AETC – Air Education and Training Command
COI – Community of Interest
AFNOSC – Air Force Nodal Operations Service Center
DISA – Defense Information System Agenc

Figure 5: *NCC Principle 4*

ment of the NCC. In Figure 4, we see a sample entry in an NCC registry. Each reference of a participant points to its entry in its own registry. The NCC registry then becomes a new, fifth type of discovery that consists of a set of relationships between participants but does not describe any individual participants in detail.

Figure 6: *NCC Principle 5*

Portfolio management of NCCs reduces the complexity of the enterprise, and forms the basis of value-based evolution of the enterprise.
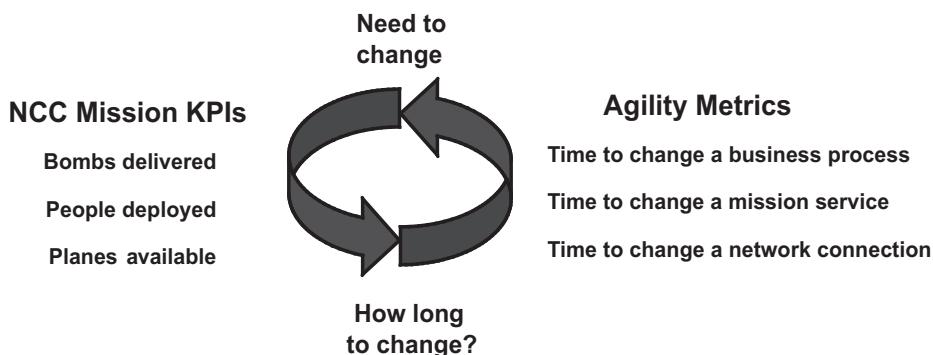
NCC Principle 3

An NCC is associated with Key Performance Indicators (KPIs). Each NCC is associated with a set of one or more KPIs which are expressed in terms of warfighter and/or business-level measurements, as appropriate. A portfolio manager can assess the contribution of the NCC to the value of the overall portfolio. Further, if the portfolio manager wants to improve the NCC value (KPI), it can be done in an objective manner and balanced against the cost of changing the NCC, as revealed by its agility profile (see Principle 4). The NCC KPIs will be derived from a combination of human/ machine observations (e.g., time to target, target assessment), as well as lower level information technology infrastructure measurements (time for certain messages to arrive or be dispatched).

NCC Principle 4

An NCC has an agility profile. One of the metrics required for managing *NCCs is the minimum time to change (including configuration)* associated with participants. These individual measurements (see Figure 5) are summarized into the minimum time to change any NCC and depend on the develop/deploy/configure processes used by each participant's organization. With quantification of agility, we can focus policy and resources on high-priority slow spots. NCC agility metrics will, for the first time, give us a numerical view of the level of agility in the enterprise, which is a key performance metric for DoD transformation.

NCC Principle 5

Portfolio Management of NCCs reduces complexity of the enterprise. NCCs are portfolio managed with KPI performance metrics balanced against agility metrics, as shown in Figure 6. This additional transformation tool helps to rationalize the process of adding net-centric capability to the enterprise. This portfolio management consists of NCCs that express multilateral relationships among participants. It assumes the participants can use network connections and services that are managed separately.
- Leadership (military and civilian) can understand the capability of an NCC because it is described in warfighter and/or business terms.
- Leadership can understand the per-

formance of an NCC because the performance is measured in warfighter/business terms.

- Leadership can understand the minimum time to change an NCC because there is an agility profile associated with it.
- Leadership can balance the need to improve KPIs against anticipated agility costs in an objective manner.

Leadership can also understand how NCCs relate to each other because NCCs can be assembled to yield compound NCCs. The set of all NCCs and their relationships to each other represent the *as-built* architecture of the enterprise, expressed in capability terms. This will focus leadership attention on integrated capability transformation and away from isolated systems or particular technologies.

Example From Global Combat Support System – Air Force (GCSS-AF)

The GCSS-AF supplies shared resources such as computing infrastructure for Air Force Operations Support applications and services. GCSS-AF provides application, hosting, data, integration, and security services, as well as the Air Force Portal. GCSS-AF integration services include an ESB that allows applications and services to exchange messages. GCSS-AF is currently developing its first NCC. Figure 7 shows how GCSS-AF assembles mission and infrastructure services into NCCs to deliver a net-centric publish/subscribe capability.

Figure 7 shows a notional example of a simple personnel notification: a key enabler of net-centric warfare that helps share awareness among participants and increases self-synchronization. Publishing Service S-1 (personnel) sends a notification message to the ESB E-1 service on the ESB Network-1 side of GCSS-AF. The notification takes the form of a publish message, M-1. The ESB E-1 then pushes Message M-1 to Subscribing Service S-2 (force readiness) on the ESB Network-1 side and also to Subscribing Service S-3 (warfighter) on the ESB Network-2 side via the cross-network service E-2 (semi-automated air gap).

Even this very simple example (one publisher, two subscribers) involves 10 participants: three services, two ESBs (ESB network-1 and ESB network-2), the cross-domain service, the message payload schema/semantics, and three
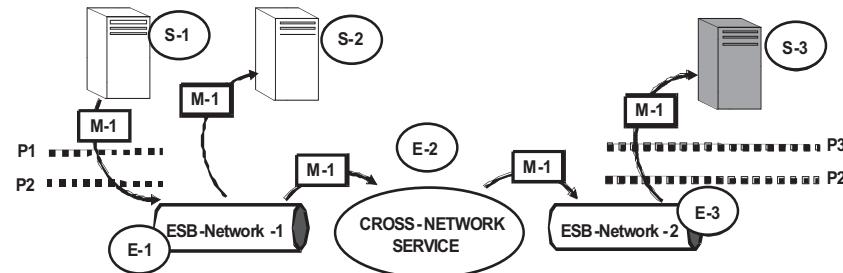
NCC Name	Users	Systems and Services	Messages	Security Perimeters	CONOPS and Doctrine	Mission Effectivess KPIs
Notify - 01	none	S-1 (personnel) S-2 (force readiness combat support) S-3 (force readiness warfighter) E-1 (ESB) E-2 (ESB) E-3 (ESB)	M-1 (airman status change)	P1 (personnel security) P2 (GCSS - AF security) P3 (warfighter security)	Alert Airman status change	1. Average time to deliver alert from personnel to warfighter 2. Percent delivered from personnel to warfighter

Figure 7: *GCSS-AF Notification Using Enterprise Services*

security perimeters. If any one of the participants must change, each of the remaining participants must be notified beforehand. Each type of change must be coordinated and must be associated with different agility metrics. Table 1 continues the notional example.

These agility metrics come into play when evaluating proposed NCC changes to determine the shortest amount of time needed to make a version change. We can balance the value of the change against cost and time.

Summary

Beyond creating NCCs in a large environment such as GCSS-AF, it is an easy extension to envision NCCs that span environments, such as GCSS-AF to GCSS-Marine (GCSS-M). The NCC would still be recorded as a single entity in one NCC registry. One of the participants could be a GCSS-M service, and the NCC entry in the NCC registry would point to the service description in the GCSS-M service registry, as well as service descriptions in GCSS-AF. The message exchanges would likely be enabled by inter-ESB connections between the two environments.

Federating ESB connections is beyond the scope of this article, but is actively explored by current GCSS-AF activities.

Additionally, NCCs will ultimately be a collaboration of the acquisition community and operational users in the field. We must be able to support both institutional pieces of an NCC, as well as be able to substitute devices on the tips of the NCC in the field by leveraging configuration and/or lightweight development.

Net-Centric Warfare and NCOs are taking place every day in support of the Global War on Terrorism. The United States and its allies in NATO continue to leverage Information Age technology to support missions worldwide. As more systems are deployed and the need for enterprise level coordination and information exchange increases, the concept of NCCs can provide a development and maintenance advantage.

NCCs do not preclude the use of Web services that are intended to be used by *everybody*, such as weather. There will continue to be a need for Web services that are put in the field,

Table 1: *Sample NCC Agility Metrics*

The ESB needs to be configured when new subscribers are added – no impact to existing subscribers, but the publisher must be notified due to security policy.	1 day
SOA lets us easily substitute services as long as interfaces are maintained – no impact to other participants.	5 days
Subscribing services need to change if message payload semantics change – potential large impact – notification and analysis required.	180 days
Cross-domain services if implemented as a guard need to change filters if message semantics change – changes in guard filters require a lengthy accreditation process.	6 months to 1 year

used relatively anonymous in a bilateral fashion (user/Web service), and will not change much over time. NCCs are intended for cases where we want repeatable, evolvable multilateral capabilities. There is no choice but to record them and manage them.

NCC's organizing principles enable us to build net-centric capabilities from relationships among new and existing information systems and users. These capabilities are measured with KPIs. They also allow us to maintain version control across all participants in an NCC and to track agility metrics, which quantify the minimum time needed to change individual NCC capabilities and the enterprise as a whole. Understanding the overall agility of the enterprise is critical to a successful transformation of the enterprise to net-centricity.◆

Note

1. A kill or supply chain links participants together in a common activity. For example, in a kill chain, we can have a targeter, an air space controller, and a pilot all working together to affect a target. In a supply chain we have the product company, distributor, shipping company, and retail stores working together to bring products to the consumer.

About the Authors

Harvey Reed joined MITRE in 2004. He is currently the chief engineer for the GCSS-AF. He led the delivery for the first ESB in the Air Force, delivered in March 2005. He is currently leading the delivery for the first Metadata Environment in the Air Force, to be delivered in the Summer of 2007. Prior to joining MITRE, Reed was a product manager at Sonic Software for business process products, as well as a voting member of the OASIS Web Services Business Process Executive Language Technical Committee. He has a bachelor's degree from Purdue University in pure math and computer science, and a master's degree from Georgia Tech University in computer and information science.

MITRE
45 Arnold ST
MS 1600H
HAFB, MA 01731
Phone: (781) 377-0823
Fax: (781) 377-1423
E-mail: hreed@mitre.org

Fred Stein is a retired U.S. Army Colonel. Currently, he is a MITRE senior engineer for Network Centric Warfare. His position and location at Ft. Hood, Texas within the Central Test and Support Facility provides him the opportunity to interface directly with the Army as it digitizes its force and moves toward the Future Combat Systems. He also serves as a member of the the Swedish National Advisor Board on Network Based Defense, advisor to the Office of Secretary of Defense Office of Force Transformation, and is an Assistant Professor at the Army War Collage. Stein has a bachelor's of science in industrial management from Georgia Technical Institute and a master's of science in operations research (business) from Boston University.

MITRE
P.O. Box 10237
Killeen, TX 76547
Phone: (254) 702-0815
Fax: (703) 564-9848
E-mail: fstein@mitre.org

CALL FOR ARTICLES
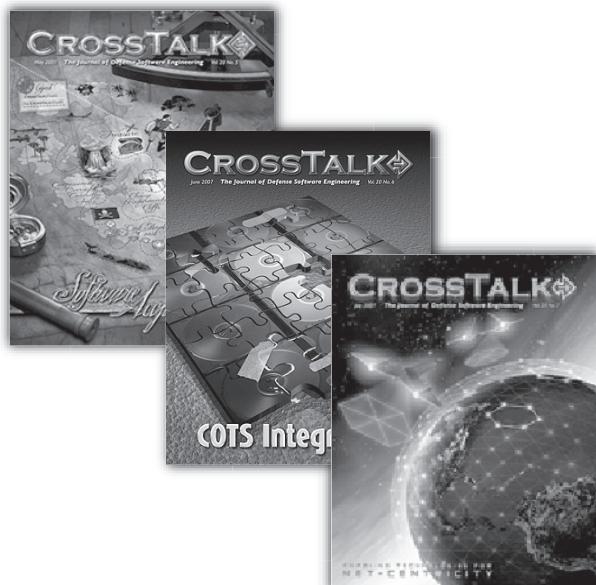
If your experience or research has produced information that could be useful to others, CROSSTALK can get the word out. We are specifically looking for articles on software-related topics to supplement upcoming theme issues. Below is the submittal schedule for three areas of emphasis we are looking for:

Distributed Software Development
March 2008
Submission Deadline: October 19, 2007

Project Tracking
April 2008
Submission Deadline: November 16, 2007

Software Safety
May 2008
Submission Deadline: December 14, 2007

Please follow the Author Guidelines for CROSSTALK, available on the Internet at <www.stsc.hill.af.mil/crosstalk>. We accept article submissions on all software-related topics at any time, along with Letters to the Editor and BACKTALK. We also provide a link to each monthly theme, giving greater detail on the types of articles we're looking for at <www.stsc.hill.af.mil/crosstalk/theme.html>.

A Unified Service Description for the Global Information Grid

Yun-Tung Lau, Ph.D.
Science Applications International Corporation

This article presents a unified approach to service description for enterprise services on the Global Information Grid (GIG). The approach introduces the concept of service module. It also identifies the links between various standards and frameworks of service description through mappings of metadata. These linkages provide end-to-end traceability for enterprise services across the architecture and design levels, thereby facilitating the development of service-oriented architectures (SOAs).

As a Department of Defense (DoD) transformational initiative, the GIG will provide a set of globally interconnected, secure end-to-end information capabilities to support operational missions conducted by various communities of interest (COIs) in the warfighting, business, and intelligence mission areas [1]. These capabilities will be fulfilled by GIG enterprise services, which are self-contained, stateless functions with well-defined interfaces that allow discovery and use of the services [2]. Such enterprise services resemble subroutines or functions in traditional computer programming except that they can be invoked by other computer programs over a network, and they are typically at a higher (mission operation) level.

SOAs are promising architecture paradigms for building GIG enterprise services. In an SOA, a set of loosely coupled services works together seamlessly and securely over a network to provide functionalities to end users [3]. As shown in Figure 1, the service provider registers information about a service interface at a service registry (step 1 in Figure 1). Service consumers can find the service from the registry (step 2) and then invoke the service through the service interface (step 3).

A typical SOA has many service consumers and service providers. The service registry may consist of a federation of registries or repositories across an enterprise. An example of an SOA on the GIG is Net-Centric Enterprise Services (NCES) [4], which provide a set of core enterprise services, including security service, service discovery, machine-to-machine messaging, and mediation for data transformation. Other applications and services on the GIG can utilize these general purpose core services to perform common functions. For COIs, enterprise services may be developed within an SOA. For example, in the command and control area, services such as blue (friendly) force location and target management services can be part of the upcoming Net-Enabled Command Capability (NECC) SOA [5].

A service description describes the way a service consumer interacts with the service provider, including the format of the request/response (messages), preconditions and post conditions, security information, quality of service (QoS) levels, etc. Some of this information is packaged into machine-readable interface contracts (e.g. Web Service Definition Language [WSDL] files). Others are entered into service registries for discovery (e.g. a Universal Description, Discovery, and Integration [UDDI] registry). Consequently, service descriptions play a central role in an SOA. They are key assets of an enterprise and should be part of the shared knowledge in the enterprise.

However, as various industry and DoD standards and frameworks of service description emerge over time, each framework tends to address a specific need without linking itself to the overall SOA engineering life cycle. This article identifies the links between existing service description standards and DoD frameworks, thereby establishing an end-to-end picture of a service and its role in an enterprise.

The following sections describe the artifacts for service definition and develop the relationships and mappings among them. I also provide a complete object model for an overall service description.

Service Definition Artifacts

Because of the central role played by service descriptions in an SOA, they are needed practically in all phases throughout an SOA engineering life cycle. At the enterprise architecture level, the DoD Architecture Framework (DoDAF) [6] is used for programs of record across the DoD. DoDAF provides the guiding principles for modeling and designing architectures in the following three views:

- The Operational View (OV) describes the tasks, activities, operational elements, and information exchanges required to accomplish missions.
- The Systems View (SV) describes systems and interconnections supporting operational functions.
- The Technical View (TV) includes technical standards, implementation conventions, rules, and criteria that

Figure 1: *Basic Interactions in a Service-Oriented Architecture*
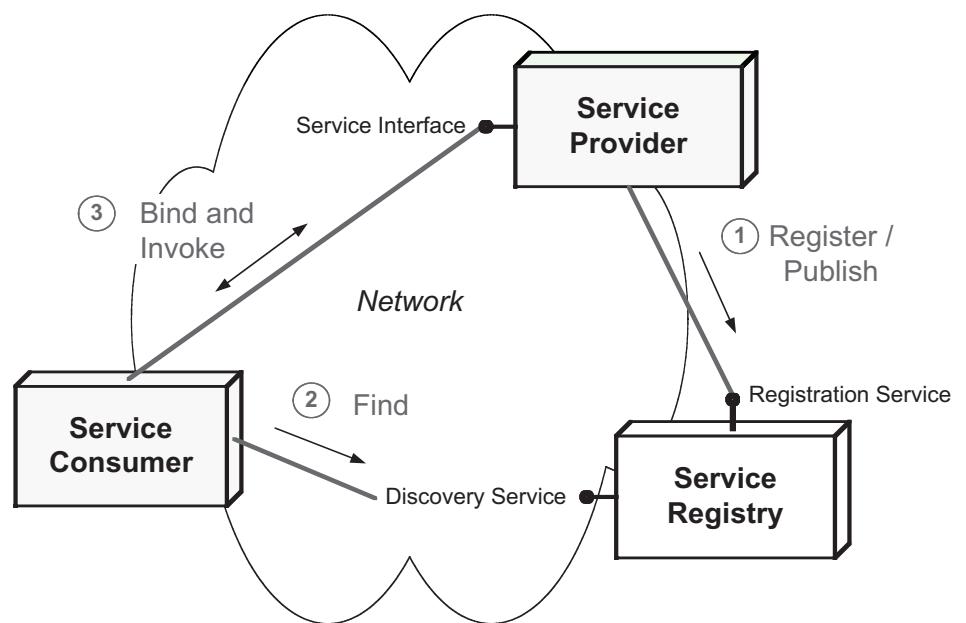
Artifacts	Representative Usages
DoDAF SV-6	Portfolio management of services. Enterprise architecture and high-level system design.
SST	Documentation of services for the GIG.
WSDL	Definition of services readable by Web service engines.
UDDI Registry	Discovery of service and provider information.
ebXML Registry	Life-cycle management and discovery of services.

Table 1: *Artifacts for Service Definition*

guide systems implementation.

Each view has a set of products. Among the SV products, the Systems Data Exchange Matrix (SV-6) specifies the characteristics of data exchange between systems. The characteristics are captured in tabular form and include data description, producer and consumer, performance attributes, security information, etc. For SOAs, similar characteristics can describe data exchange between service consumers and service providers. One may therefore apply the SV-6 product to service descriptions at the architecture level. In this case, the producer in the SV-6 matrix represents the service provider and the consumer represents the service consumer.

At the design and implementation level, the Service Specification Template (SST) has been proposed as part of the GIG Net-Centric Implementation Document series [7]. The SST identifies a set of elements (grouped by categories and subcategories) that describe a GIG enterprise service. These elements indicate what the service does, how to access the service, the security mechanisms or restrictions for the service, relevant performance information, etc. The SST is intended to aid in the specification, implementation, documentation, and discovery of services across the GIG.

For implementation and deployment, several industry standards are widely used: WSDL for machine-readable interface contracts [8] and UDDI [9] and Electronic Business eXtensible Markup Language (ebXML) registries [10] for discovery of services. They contain different aspects of information about the services. Table 1 gives a summary of the above artifacts for service definition.

Unified Service Definition

These artifacts were developed separately for the uses shown in Table 1. To gain a deeper understanding of the relative roles they play in building an SOA, one must establish an end-to-end linkage across them. We can achieve this in two steps. First, we introduce the concept of a service module in order to facilitate the transition from architecture to design of an SOA. Second, we identify the mappings of metadata between the artifacts.

Service Module

A key activity in the early stage of SOA development is identifying the services. These services, identified at the enterprise architecture level, are often (though not always) *service modules*, which handle operational processes in a certain mission area. Each service module may contain multiple concrete services which are implemented

in the design and development phases and invoked by consumers after deployment.

As an example, a task management service module may contain two concrete services: retrieval and administration. The task retrieval service is consumed by general users assigned to perform the tasks. The task administration service, on the other hand, is used by administrative users who set up and maintain the tasks. An enterprise architecture artifact, such as the SV-6, may capture information about the task management service module only, or it may also contain information about the two concrete services.

Figure 2 gives the relationship of service and service module in Unified Modeling Language (UML) notations [11]. It shows that a service module may contain multiple child modules, as indicated by the asterisk next to the label *Children*. A service module may be related to multiple rows in the SV-6 matrix (each row corresponding to a data exchange object in Figure 2). These rows represent data exchanges of concrete services under that service module. Alternatively, one may roll up the information from the concrete services under a module to a single row in the SV-6 matrix. Also, the notation 0..1 in Figure 2 indicates zero or one instance of an object, whereas 1 means exactly one instance. For example, a service may be associated with zero or one data exchange, whereas a data exchange is always associated with one service (under SOA).

Depending on the level of details conveyed by an enterprise architecture, one may provide data exchange information at the concrete service level. In this case, a row in the SV-6 matrix contains information about an individual concrete service. The label {xor} in Figure 2 indicates that a row in SV-6 may be associated with either a concrete service or a service module, but not both. In what follows, concrete services at the design level are simply

Figure 2: *Relationship of Service and Service Module*
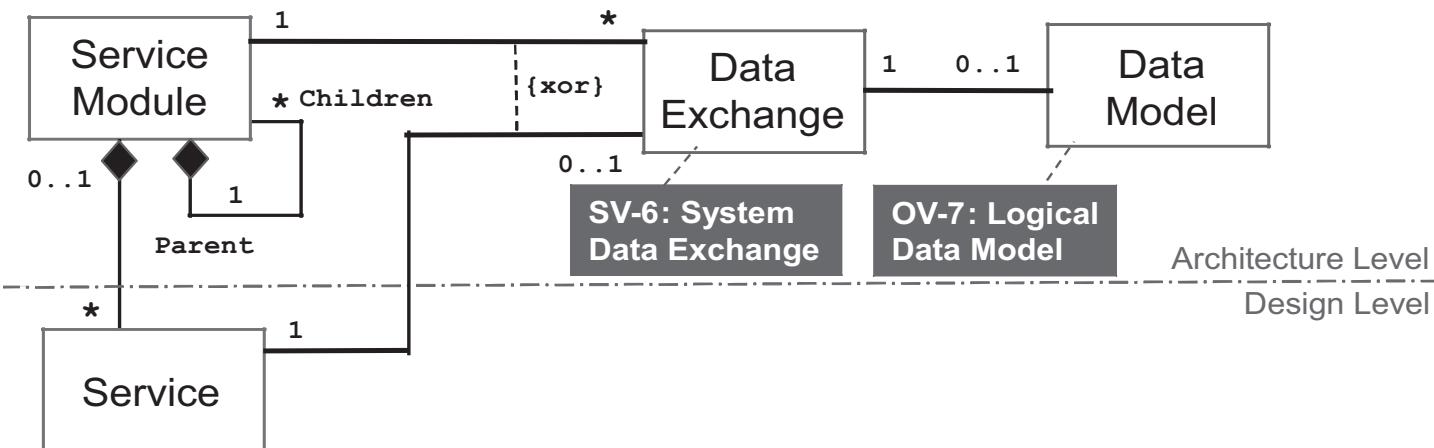
called services.

Figure 2 also shows that a data exchange (a row in the SV-6 matrix) may be associated with zero or one logical data model (OV-7). Identifying data models at the architecture level helps promote sharing of data across services, which is a key tenet of SOAs.

Mapping of Metadata

Service definition is about information that describes a service. In other words, it contains metadata about a service. One can group those metadata into categories, such as security information, service level information, etc. Different artifacts for service definition focus on different categories of metadata. By mapping the metadata across the artifacts, one establishes the linkages between the artifacts.

Table 2 gives the mappings of metadata across the artifacts for service definition. An empty cell indicates that there are no corresponding metadata for that artifact. For example, SV-6 does not carry version information, which is needed for design and implementation. Note that if an entry in the SV-6 represents a service module, then there is no corresponding mapping to the other artifacts. This is because those other artifacts are below the architecture level.

In the SV-6 matrix, the parent-child relationship can be indicated by a dot-delimited System Interface Identifier in the form of x.y.z…, where x, y, z are integers. For example, the following shows a Security Service Module and a partial list of services under it:
- 1.7   Security Service Module.
- 1.7.1 Certificate Validation Service.
- 1.7.2 Policy Decision Service.
- 1.7.3 Policy Retrieval Service.

Artifacts at the design level usually do not carry information on such a parent-child relationship.

Table 2 shows, other than architecture level information, the SST provides rather comprehensive information about a service. The information needed for invoking a service is mapped to WSDL, whereas the information for discovery of services is mapped to a UDDI or ebXML registry.

UDDI uses tModel (which basically contains name-value pairs) to facilitate searching by attribute values. The mapping strategy in this case is to link the elements in SST to a UDDI tModel. For example, an *InformationSecurityMarking* element under the Service Information/Security category maps to an *InformationSecurityMarking* tModel. For an ebXML registry, *ClassificationScheme* and *ClassificationNode* are the equivalent of a tModel. One may therefore

	DoDAF SV-6	SST	WSDL	UDDI Registry	ebXML Registry
Service Name	√	√	√	√	√
Version		√	√	√	√
Parent/Child Relation	√				
Transaction Information	√				
Data Standard	√				
Namespace		√	√	√ [1]	√ [2]
Operations		√	√	√ [1]	√ [2]
Access Point		√	√	√ [1]	√ [2]
Security Information	√	√		√ [1]	√ [2]
Service Level Information	√	√		√ [1]	√ [2]
Schedule Information		√		√ [1]	√ [2]
Contact Information		√		√ [1]	√ [2]

Note 1: Each of these categories corresponds to a UDDI tModel, which basically contains name-value pairs.
Note 2: Each of these categories corresponds to an ebXML ClassificationScheme.

Table 2: *Mapping of Metadata by Category*

construct the mapping similarly.

Detailed UML Model and Mapping

The material in this section is intended for SOA practitioners who would like to find out details of the mappings described in this article. They may further use the Web examples[3] as references for building service definitions in an SOA.

Figure 3 presents a full UML model for Service Module and Service. In addition to the relationship given in Figure 2, it shows the linkages from Service to SST, WSDL, and other related artifacts at the design level. Here the UML notation 0..1

indicates zero or one instance of an object. An asterisk represents zero or more instances, whereas 1 means exactly one instance. The label {xor} indicates that a Data Exchange object (a row in SV-6) may be associated with either a concrete service or a service module, but not both.

The mappings at the field or XML element level between these artifacts are given in the spreadsheet USD_Mapping _and_Example.xls[3].

For the SST, an earlier version (v. 2.0) of the document defines an XML schema, which is called the Service Definition Framework (SDF). The sample XML data is based on that SDF schema and is in the

Figure 3: *UML Model for Service Module and Service*
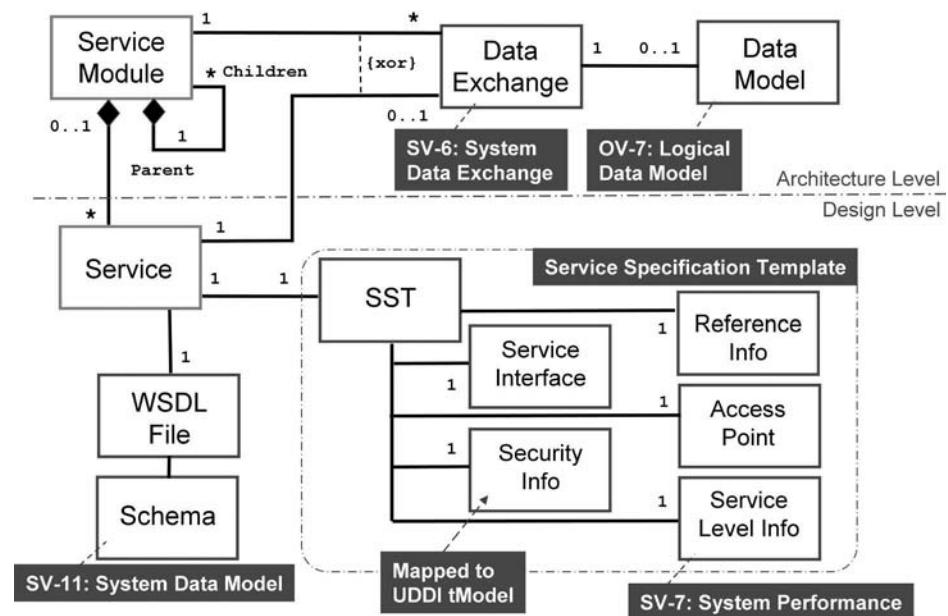
file CES_Security_CVS(SDF). They may be useful as references for building service definitions.

Finally, in the spreadsheet, we use the following XPath notations in identifying elements in XML data for the mappings:
1. /A/B/C: Element C under element B, which is under the root element A.
2. D/E[@x]: Attribute x of element E under element D.

For example, the XPath expression

/SDF/ServiceAccessPointInformation/ ServiceAccessPoint/operation

corresponds to "getStatus" in the XML data below:

```
<SDF>
    ...
    <ServiceAccessPointInformation>
        ...
        <ServiceAccessPoint>
            <operation>getStatus
            </operation>
            <binding>SOAP/HTTP
            </binding>
            <port>http://decc2.dod.
            mil/CES/Security/CVS
            </port>
            <POCIndex>Jane Smith
            </POCIndex>
            <SupplementalInformation>
            OCONUS
             </Supplemental
            Information>
        </ServiceAccessPoint>
    </ServiceAccessPointInformation>
</SDF>
```

Similarly, the XPath expression

/definitions/binding/operation[@name]

points to the operation name "getStatus" in the XML data below:

```
<definitions>
  ...
  <binding name="CertificateValidation
    ServiceSOAPBinding" ... >
    <soap:binding style="document"trans
     port="http://schemas.xmlsoap.org/
     soap/http"/>
    <operation name="getStatus">
    ...
     </operation>
  </binding>
</definitions>
```

Closing Remarks

The introduction of service module enables a unified approach for service definition across the architecture and design levels. When performing a top-down SOA design, one may start with one or more services in a module and later refine them into more services. The service module remains the same during this refinement, therefore allowing the design to evolve without affecting artifacts at the architecture level.

On the other hand, in a bottom-up approach, one can map design level information in the SST to WSDL, UDDI, and ebXML, as shown in Table 2. As one refines the individual services, one may further group related services into service module at the architecture level.

The unified service description thus lends flexibility to the system engineering process and provides end-to-end traceability for enterprise services in the GIG. Even though one may use different tools for the different standards and frameworks in Table 2, those tools can in principle be integrated or linked together to provide a complete picture of the services.◆

References[1]

1. "Global Information Grid Capstone Requirements Document." JROCM 134-01. U.S. Joint Forces Command, 2001 <https://jdl.jwfc.jfcom.mil>.
2. United States. Assistant Secretary of Defense, Networks and Information Integration. "Net-Centric Operations and Warfare (NCOW) Reference Model." v1.1. Washington: DoD, 2005 <https://standmgt.disa.mil/restricted/ncow.html>.
3. Erl, Thomas. "Service-Oriented Architecture (SOA): Concepts, Technology, and Design." Prentice Hall, 2005.
4. United States. Defense Information Systems Agency (DISA). Initial Capabilities Document for Global Information Grid Enterprise Services." DISA, 2004 <https://gesportal.dod.mil>.
5. "Net-Enabled Command Capability Milestone A Acquisition Decision Memorandum, Assistant Secretary of Defense, Networks and Information Integration." Washington: DoD, 2006 <https://gesportal.dod.mil/sites/necc>.
6. United States. Assistant Secretary of Defense. "DoD Architecture Framework." Ver. 1. Washington: DoD, 2007.
7. United States. Assistant Secretary of Defense. "Global Information Grid Net-Centric Implementation Document: Service Specification Template (S300)." Ver. 4.0. Washington: DoD, 2006 <https://gesportal.dod.mil>[2].
8. Christensen, Erik, Francisco Curbera, Greg Meredith, and Sanjiva Weerawar-ana. "Web Services Description Language." 1.1 World Wide Web Consortium, 2001 <www.w3.org/TR/wsdl>.
9. "Universal Description, Discovery, and Integration (UDDI)." v3.0.2. Organization for the Advancement of Structured Information Standards, 2004 <http://uddi.org/pubs/uddi-v3.0.2-20041019.pdf>.
10. ebXML. Organization for the Advancement of Structured Information Standards <www.ebxml.org/specs>.
11. International Organization for Standardization/International Electrotechnical Commission "Unified Modeling Language." 19501. International Organization for Standardization. <www.iso.org>.

Notes

1. Some Web sites quoted here require a user account for access. Online application forms can be found on the sites. Some require government sponsorship.
2. An XML schema was included in version 2.0 of this document, titled Service Definition Framework. The sample XML are based on that schema.
3. The example files are available for download in the online version of this article.

About the Author

Yun-Tung Lau, Ph.D. is Vice President of Technology at Science Applications International Corporation (SAIC). He has been involved in large-scale software architecture, design, and development for 18 years. Lau has served as chief architect for many software and enterprise architecture projects, from scientific computing and electronic commerce to command and control systems. He holds a master of technology management. Lau has published many articles in professional journals, and authored the book *The Art of Objects: Object-Oriented Design and Architecture*.

SAIC
5113 Leesburg Pike
STE 200
McLean, VA 22041
Phone: (703) 824-5817
Fax: (703) 824-5836
E-mail: yun-tung.lau@saic.com

Beyond Defect Removal: Latent Defect Estimation With Capture-Recapture Method

Joe Schofield
Sandia National Laboratories

Defect removal and defect prevention techniques are no longer good enough to inspire confidence in software products. Techniques that help predict the number of remaining defects in software products can further boost customer confidence. Such techniques are easy to perform and have been used outside the realm of software engineering to produce reliable estimates for decades in the area of animal, bird, fish, and insect counts, and more recently for estimating the prevalence of the Severe Acute Respiratory Syndrome and cancer occurrences. This article describes the business case for removing defects and demonstrates how the usage of the Capture-Recapture Method (CRM) in defect removal activities can predict the number of estimated defects remaining in a product. This estimate can then be used to make quantified, data-driven decisions on how to proceed with a software product.

In December of 2005, Ford, Marriott, Sam's Club, and the Justice Department were all vilified in a nationally recognized information magazine for having customer data compromised through either theft or their inability to secure sensitive data [1]. Medical staff report that 770,000 medication mistakes occur each year in the U.S.; these errors are more than penmanship issues, transcription, data entry, and other preventable errors [2]. In 2004, interface issues between Hewlett-Packard's order entry system and SAP AG systems triggered $40 million in lost revenues [3]. Early in 2006, a property in Indiana valued at $121,900 had its value assessed for tax purposes at $400 million. The common thread to each of these incidents is software defects.

As recently as 2003, less than one-third of software organizations had a quality assurance group or processes [4]. Software developers like to use phrases like *level of rigor* and *quality commensurate with risk* to avoid or minimize the need for investing time in the quality of their products. Sound familiar? Tell the victims of the defects that it is *just a computer problem*, a *glitch*, an *issue*, a *foul-up*, a *snafu*, or a *bug*. Are they feeling better yet? What do you think is the level of confidence these victims have in the supplier? Will these consumers return and advocate the products and services they purchased?

Driving down the street we notice how credentialed the rest of our world has become. Attorneys, accountants, financial planners, physicians, surgeons, nurses, plumbers, electricians, engineers, and mechanics – they are all certified. But anyone with some level of educational or experiential hacking can write code. Credentials do not eliminate defects; verify this with a *certified* attorney. Credentials do however offer a measure of confidence to the consumer that the holder of the certification is trained and tested in the use

of some body of knowledge, and often, subscribe to some code of ethics.

In lieu of certification credentials, another approach to raising the confidence of *software* consumers is to embrace defect removal and prediction techniques. The latent defect derivations that result from the prediction techniques are not rocket science. A peer recently taught fifth

> **"CRM affords a product development team the opportunity to employ statistical approaches to verify the goodness of a product as it is designed, developed, and deployed."**

graders how to perform defect prediction; they became quite familiar with those techniques in merely a few hours.

Defects found during testing reveal as much about the adequacy of the process as they do the quality of the product. Is it not it an ominous sign when companies advertise that they are looking for more software testers? Clearly, quality (Q) without defect removal (Dr) is just faking (F) it (Q – Dr = F). But is the removal of *identifiable* defects adequate?

CRM affords a product development team the opportunity to employ statistical approaches to verify the *goodness* of a product as it is designed, developed, and deployed. Defect removal is woefully late and excessively costly during test (and even more so after release). CRM can be

used by product teams to validate requirements and verify design criteria to reduce latent defects by estimating how many defects persist in their products. With this data, teams can make objective choices about proceeding or spending additional time to address unfound but predicted defects in their products. Eventually, practitioners benefit from the assurance of knowing that their products meet the expectations imposed upon them. Management benefits from the increased confidence that latent and hidden costs of post-delivery fixes are predictable, understood, and controlled. Ultimately, estimated latent defect data reduces the *risk* in risk management.

This article is not just another prognostication about a defect-induced apocalypse, nor is it another article to encourage more thorough testing to remove defects; after all, defect removal by testing is too similar to inspecting quality into a product as it rolls off the production line. This article is not about the effectiveness of inspections and peer reviews to remove defects close to their point of injection. So what, you might patiently ponder, *is* the purpose of this article? Not so fast.

Recently, a mid-level executive proudly shared that his team had just completed a one million line of code (1 MLOC) project with only 40 *issues* (notice the euphemism) reported. Ignoring the misunderstanding on his part regarding the significance of the size of the product [5], let us focus on the defects (issues) per MLOC. Forty deaths per million air miles or 40 injuries per million air passengers would not be acceptable to consumer safety groups. Forty deaths per year from providing wrong prescriptions is not healthy (the actual number is 7,000 per year) [6]. Forty cruise passengers returned to the wrong debarkation port would not float either. So why would 40 *issues* with a software delivery be hailed as laudable?

Does this statement reflect more about the expectations we have for software products or the state of maturity of software development in general?

While possibly more troubling or sensational, the above examples do provide perspective into the serious nature of defects of any kind. Incidentally, the 40-issue-defect-product above was a highly sensitive data collection system.

The lingering question in my mind was *how many defects have you and your customer not found, yet?* I knew he did not know, and I hardly wanted to ruin his otherwise sunny day.

So what is the purpose of this article? Simply stated, it is to encourage software engineers to use predictive techniques for determining the quality of products throughout their product development activities. The CRM is one such technique.

Brief Background
Our organization received a Capability Maturity Model for Software Level 3 certification in 2005. We rely on Personal Software Process℠ and Team Software Process℠ (TSP) as enablers of practice improvement. A colleague, Tom Cuyler, recently received his TSP certification. For the past year the organization has been re-engineering its software processes with a CMMI® Maturity Level 4 target. As part of our ongoing process improvement, Cuyler suggested we consider using the CRM which Watts Humphrey advocates in his TSP material [7]. Cuyler and I experimented with the CRM, he in his TSP work and I in our organization training.

We have collected defect data for the last five years. We know where our reported defects are injected, where they are detected, the defect type, its severity, the cost to repair, and the cost to discover (this last value is derived at a macro level). We derive and share defect leakage measures with project and management teams. We can estimate defects by function points in development and latent defects in delivered products. (Note: Latent defects can be estimated by defects reported by the customer after delivery using historical data from earlier projects. The defects not yet found by the customer, and perhaps never to be found remain unknown.)

So What's the Problem?
Defect riddled products continue to be released hindering the customer and casting a shadow of suspicion on the credibility of the supplier. Testing has not been effective in eliminating defects. Peer reviews and inspections have been effective in reducing, but not eliminating defects. Code testing tools cannot identify defects in the elicitation of requirements.

> *"Simply stated, [this article] is to encourage software engineers to use predictive techniques for determining the quality of products throughout their product development activities."*

To elaborate briefly, managers and project leaders have false confidence in product quality due to a paucity of the use of estimated latent defects in delivered products. In lieu of an approach like the CRM and statistical latent defect estimating (versus experiential or defect estimation based on *reported* defects), any claim about the quality of software is no more objective than that assertion from the aforementioned executive who deserved vigorous cross-examination.

And What's a Solution?
The CRM has been used for decades for sampling and estimating in disciplines unrelated to software engineering [8]. Even exploring the fine print and limitations of the technique, CRM is quite appropriate for peer reviews, for instance, (and even testing [if you must]). Caution: do not limit the use of CRM to peer reviews of code. Peer reviews and stakeholder reviews are useful mechanisms for verification and validation early in requirements capture, through design, as well as later during construction and testing. Here's a simple example of applying the CRM to a product that is being peer reviewed.

In Table 1, three product engineers identified a total of seven defects in a product; these are identified in the Defect Number column. In the next three columns, we associate which defects were found by which engineer in their individual preparation for the peer review. In Column A, the defects by the engineer who found the most unique defects are identified. In this case, Larry found the most unique defects, and Column A duplicates Larry's findings. In Column B, each defect that was found by all of the other participants is identified. In this case, the defects found by Curly and Moe are identified. In Column C, each defect that was found in both Column A and Column B are identified (e.g., the intersection of these two columns). The counts for Columns A, B, and C are totaled in this example, 5, 4, and 2, respectively.

The CRM indicates that the estimated number of probable defects in the product is:

$$(A * B) / C$$

in the example this value is:

$$(5 * 4) / 2 \text{ or } 10$$

The CRM also indicates that the number of defects found by the participants is:

$$A + B - C$$

In the example this value is calculated as:

$$5 + 4 - 2 \text{ or } 7$$

Finally, the CRM indicates that the estimated number of defects remaining is the difference between the probable number of defects (10) and the found defects (7) or 3. The *long hand* for this calculation is:

$$((A * B) / C) - (A + B - C)$$

For our example:

$$((5 * 4) / 2) - (5 + 4 - 2), \text{ i.e., } 3$$

Table 1: *CRM Example*

Defect Number	Engineer Larry	Engineer Curly	Engineer Moe	"Column A"	"Column B"	"Column C"
1	✓			✓		
2	✓			✓		
3			✓		✓	
4	✓	✓		✓	✓	✓
5	✓			✓		
6	✓		✓	✓	✓	✓
7		✓			✓	
Totals	5	2	2	5	4	2

Therefore, in this example, the team has estimated that 70 percent of the defects in the product were identified as part of the peer review (and were/or will be removed), and that 30 percent of those defects remain.

Four important points are rendered here (The parenthetical references to CMMI are the most obvious mappings to the model and are not intended to be exhaustive.):

- First, the team has a quantified and objective process for determining the outcome of the peer review: repeat the review, accept the results of the review, or something else (CMMI Process Areas – Measurement and Analysis and Verification are supported with the CRM).
- Second, the team has an opportunity to establish defect removal thresholds – and manage to them. These thresholds could correspond to quality objectives for the organization and the project (CMMI Process Areas – Organizational Process Performance, Project Monitoring and Control, and Generic Practice 3.2 – Collect Improvement Information).
- Third, the estimated number of latent defects can be used to assess, analyze, and mitigate project risks (CMMI Process Area – Risk Management).
- Fourth, the outcome of any defect analysis can be used for improved training activities (CMMI Generic Practice 2.5 – Train People).

At a recent New Mexico Software Process Improvement Network (SPIN) meeting, Jerry Weinberg (the real Jerry Weinberg) was speaking about writing [9]. He referred to a manuscript which he had distributed to several associates. Weinberg indicated that he used the typos they reported to him to estimate the remaining typos in his document. I asked him if he used the CRM to do this, to which he responded (only slightly surprised by the question) *yes*. His writing project, in this case a book, was completed decades ago. Regrettably, the years erode the lessons and wisdom of the past.

Conclusion

CRM is widely used outside the software engineering world, and I suggest it is desperately needed inside the software engineering practices world. Easy, effective, and economical, we have found the CRM a valuable technique for quantifying confidence in products delivered. Stay tuned.◆

Acknowledgements

Thanks to Watts Humphrey for promoting this concept, to Tom Cuyler for introducing CRM in our community, to Jerry Weinberg for confirming its use outside of our initial research, and to Anna Nusbaum for her insightful review and edits of this article.

References

1. "December Data Exposures." Informationweek Feb. 2006: 19.
2. "Medication Systems." CIO June (2005): 28.
3. Levinson, Meredith. "The High Cost of Flawed Testing." CIO Nov. (2005): 66.
4. Surmacz, Joe. "Why Software Quality Stinks." CIO Dec. (2003).
5. "The Statistically Unreliable Nature of Lines of Code." CROSSTALK, Apr. 2005: 29-33 <www.stsc.hill.af.mil/crosstalk/2005/04/index.html>.
6. "When Did Six Sigma Stop Being a Statistical Measure?" CROSSTALK, Apr. 2006 <www.stsc.hill.af.mil/crosstalk/2006/04/index.html>.
7. Humphrey, Watts. Introduction to the Team Software Process. Addison-Wesley Professional, 2000.
8. LaPorte, R.E., D.J. McCarty., E.S. Tull, and N. Tajima. "Counting Birds, Bees, and NCDs." Lancet 1992.
9. Gerald Weinberg. <www.geraldmweinberg.com>.

About the Author

Joe Schofield is a distinguished member of the technical staff at Sandia National Laboratories. He is a Software Engineering Institute author and instructor for CMMI Introduction, is a trained Lean Six Sigma Black Belt, chairs the organization's software engineering process group, and leads the organization's movement from SW-CMM 3 to CMMI Maturity Level 4. Schofield chairs the Management Reporting Committee for International Function Point users Group, is active in the local SPIN, and has taught graduate level software engineering classes since 1990.

Sandia National Laboratories
MS 0661
Albuquerque, NM 87185
Phone: (505) 844-7977
Fax: (505) 844-2018
E-mail: jrschof@sandia.gov

WEB SITES

Data and Analysis Center for Software (DACS)

https://thedacs.com

The DACS is a Department of Defense (DoD) Information Analysis Center (IAC). The DACS has been designated as the DoD Software Information Clearinghouse, serving as an authoritative source for state-of-the-art software information and providing technical support for the software community. The DACS technical area of focus is Software Technology and Software Engineering, in its broadest sense. The DACS is a central distribution hub for software technology information sources. The DACS offers a wide-variety of technical services designed to support the development, testing, validation, and transitioning of Software Engineering technology. The DACS is administratively managed by the Defense Technical Information Center under the DoD IAC Program. The DACS is technically managed by Air Force Research Laboratory - Information Directorate. ITT Corporation manages and operates the DACS, serving as a centralized source for current, available data and information concerning Software Engineering and Software Technology.

Construx' Software Development Best Practice Conversations

http://forums.construx.com/

At the Construx's Software Development Best Practice Conversations forum you will find in-depth discussions of requirements, management, design, coding, testing and other software development topics. The menu bar gives you the most important links. You can: visit blogs and software best practices discussion forum, join a discussion group, and share your take on software development best practices.

Organization for the Advancement of Structured Information Standards (OASIS)

www.oasis-open.org

OASIS is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. The consortium produces more Web services standards than any other organization along with standards for security, e-business, and standardization efforts in the public sector and for application-specific markets. Founded in 1993, OASIS has more than 5,000 participants representing over 600 organizations and individual members in 100 countries. The OASIS Web site provides multiple links, memberships, and access to newsletters.

Defend America

www.defendamerica.mil

DefendAmerica, an official DoD Web site, was launched just weeks after the Sept. 11, 2001 terrorist attacks to keep the public informed about efforts by the United States and its coalition partners to combat global terrorism. The site offers the latest news, photographs, transcripts and other information about the U.S.-led war on terrorism. It highlights the words and activities of key U.S., DoD, and coalition officials related to terrorism. But DefendAmerica also offers something not so readily available in the mainstream media: daily news reports and photographs by U.S. military photojournalists on the frontlines as well as in supporting units. The site reports on the roles all branches of the military play in the war on terror: Army, Navy, Air Force, Marine Corps, and Coast Guard, those on active duty as well as in the National Guard and Reserve. It covers contributions by coalition partners who have joined the United States in the war on terror. DefendAmerica also highlights a critical but often overlooked partner in the terror war: the American public that stands by to support the troops as they take a stand against the forces of terrorism.

Free Management Library

www.managementhelp.org

The library provides easy-to-access, clutter-free, comprehensive resources regarding the leadership and management of yourself, other individuals, groups and organizations. Content is relevant to the vast majority of people, whether they are in large or small, for-profit or nonprofit organizations. Over the past 10 years, the library has grown to be a large, well-organized collection of these types of resources. There are approximately 650 topics in the library, spanning 5,000 links. Topics include the most important practices to start, develop, operate, evaluate and resolve problems in for-profit and nonprofit organizations. Each topic has additionally recommended books and related library topics. As much as possible, library administrators attempt to focus content on easy-to-apply, general information that will be of use to anyone when managing themselves, other individuals, groups and organizations.

Web Services and Service-Oriented Architectures

www.service-architecture.com

This site will help you get started with Web Services and service-oriented architectures. It features free articles, services, and product listings that can be used to develop a service-oriented architecture using Web Services. Online articles provide an extensive overview of Web Services, related standards, and technologies that can be used in service-oriented architectures. There are nearly 400 pages of articles. Services help your organization decide how to use Web Services in a service-oriented architecture. Product listings connect you to the vendor sites for each of the technologies. The online articles section provides an extensive overview of Web Services, related standards, and technologies that can be used in service-oriented architectures. Web Services make up a connection technology. It is a way to connect services together into a service-oriented architecture.

Software Program Managers Network (SPMN)

www.spmn.com

The mission of the SMPN is to identify proven industry and government software best practices and convey them to managers of large-scale software-intensive acquisition programs. Applying extensive *in the trenches* experience, the SPMN enables program managers to achieve project success and deliver quality systems on schedule and on budget. To date, more than 250 DoD programs have benefited directly from SPMN expert consulting.

Common Threads in Life

A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.

–Robert A. Heinlein, *Time Enough For Love* [1]

Looking for patterns, themes, and repeated motifs is a common technique for understanding many subjects. Gamma [2] did a great job describing the most proven software development patterns. The Software Program Managers Network (SPMN) [3] has captured dozens of lessons learned in managing projects. But like the late Robert Heinlein, my focus is a bit broader than those examples.

Fifteen themes and patterns have emerged (so far!) from wildly disparate activities. Fields as diverse as dancing, project management, golf, engineering, and massage therapy have contributed to this collection of observations about how we think, plan, move, and analyze. Here are 11 of 15 observations most relevant to software engineering and project management.

1. **Focus beyond what is possible or seems immediately relevant**. Golf, tennis: Continue the swing smoothly long past contact with the ball.
   Karate: Focus the target of a technique farther than you will actually be able to strike.
   Dance: Focus your attention and maintain connection beyond your body – into the earth, into the sky, with your partner, to the audience.
2. **Focus attention on the desired outcome, not what you're avoiding.**
   Golf: The best way to hit the ball poorly is to focus on what you don't want the ball to do.
   Project management: A project plan identifies the tasks needed to achieve the desired objective of the project.
   Karate: Assume you will be successful and determine how to make it happen.
3. **Recognize risks without dwelling on them.**
   Project management: sound risk management is critical to success, but can't be the only activity.
   Golf: plan for likely errors, without falling into the trap of the previous observation.
   Karate: Recognize your opponent's strengths, yet plan your strategy for success in spite of them.
4. **When something goes well, stick to the basics.**
   Project estimation, karate, golf, piano, dance: Practicing the basics is the key to achieving better performance.
5. **When something goes badly, go back to the basics.**
   Project management: A classic mistake is to abandon the project plan when something goes wrong – instead, that's the time to return to basic understanding of tasks to be done and measuring progress toward achieving them [4].
   Golf: After a bad shot, the best way to avoid a string of more bad shots is to focus on basic technique.
6. **Follow by rote at first; then with experience, tailor your approach.**

Project management, dance, karate, massage, construction, etc: When first learning a new skill, it is common to follow literally a prescribed set of actions. As you develop more skills, you develop the ability to adopt and blend techniques from a variety of sources.

7. **Attention to detail separates good from great.**
   Carpentry, fashion, music, dance: The difference between ordinary work and excellent work is often in attention to details.
8. **Balance similarity and opposites.**
   Project management, engineering: Most management and design decisions involve balancing conflicting needs (speed vs. quality, light vs. strong, etc.), yet good design practice encourages reuse and application of patterns.
   Dance, music: Use changes between fast and slow, smooth and sudden, harmony and dissonance, symmetry and asymmetry, and repetition and novelty to create interesting work.
9. **Left field is a good place to visit often.**
   Program management: Successful contract approaches often use unconventional structure.
   Engineering, science: Many great insights have come from pulling together seemingly unrelated concepts and discovering synergy among them.
10. **All things are rarely equal.**
    Risk management: Risks need to be quantified to see which are most important.
    Software engineering: need to tailor the scope of testing to match the complexity and criticality of the component.
    System modeling: need to model the critical aspects of a system and set aside the rest.
11. **Examine a problem with different sets of eyes.**
    Requirements engineering: Get input from all kinds of users and stakeholders.
    Astronomy: Insights have been obtained from looking the same direction, but using visible, infrared, or polarized light, or using a radio telescope.
    Bioengineering: Learn about a material by correlating different aspects of it – physical properties, chemical reactions, luminescence, and radioactivity.

The other four observations relate mostly to physical movement and are in the online version of this document <www.stsc.hill.af.mil/crosstalk/2007/08/0708/backtalk.html>. This a starting point for continued thought and evaluation – feedback, rebuttal, and additions are welcome!

— Glenn Booker
gbooker@drexel.edu

References

1. Heinlein, Robert A. Time Enough for Love: The Lives of Lazarus Long. New York: Ace, 1994.
2. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns, Elements of Reusable Object-Oriented Software. Indianapolis: Pearson, 1995.
3. SPMN Guidebooks. Integrated Computer Engineering. 2006 <www.spmn.com/pdf_download.asp>.
4. McConnell, Steve. "Rapid Development." Redmond, WA: Microsoft Press, 1996.

Building Solutions for the Systems of the Past, Present, and Future!

CMMI Level 5

309TH SOFTWARE MAINTENANCE GROUP

AS9100   ISO 9001

Please contact us today

Ogden Air Logistics Center
309th Software Maintenance Group
(formerly MAS Software Maintenance Division)
Hill Air Force Base, Utah 84056

Commercial: (801) 777-2615, DSN 777-2615
E-mail: ooalc.masinfo@hill.af.mil
or visit our website: www.mas.hill.af.mil

CrossTalk **/ 517 SMXS/MXDEA**
6022 Fir AVE
BLDG 1238
Hill AFB, UT 84056-5820

PRSRT STD
U.S. POSTAGE PAID
Albuquerque, NM
Permit 737